

09-01-00

A

08/31/00
JCS83 U.S. PTO

10/06/02 U.S. PTO
09/652748
08/31/00

Patent Application Transmittal

(only for new nonprovisional applications under 37 C.F.R. 1.53(b))

Correspondence Address:
FROMMER LAWRENCE & HAUG LLP
745 FIFTH AVENUE
NEW YORK, NEW YORK 10151
TEL: (212) 588-0800
FAX: (212) 588-0500

Date: August 31, 2000
Attorney Docket No.: 450117-4278.2

ASSISTANT COMMISSIONER FOR PATENTS
Box Patent Application
Washington, D.C. 20231

Sir:

With reference to the filing in the United States Patent and Trademark Office
of an application for patent in the name(s) of:

PAUL HODGINS, GERT JOSEF ELISA COPEJANS, YOERI APTS, JOHAN DE VOS
entitled:

METHOD AND APPARATUS FOR SERVING DATA

X Continuing Application

 Continuation X Divisional Continuation-in-Part (CIP)
of prior application serial no. 08/979,474, filed November 26, 1997.

[Note: If priority under 35 U.S.C. 120 involves a series of respectively copending
applications, then in this amendment identify each and its relationship to its immediate
predecessor.]

X The prior application is assigned of record to SONY EUROPA B.V.

The following are enclosed:

X Specification (32 pages)

X 13 Sheet(s) of Drawings

X 12 Claim(s) (including 2 independent claim(s))

 This application contains a multiple dependent claim

X Our check for \$ 690.00, calculated on the basis of the claims
existing in the prior application (less any claims canceled herein) as
amended by any enclosed preliminary amendment as follows:

Basic Fee, \$690.00 (\$345.00)	\$ 690.00
Number of Claims in excess of 20 at \$18.00 (\$9.00) each:	-0-
Number of Independent Claims in excess of 3 at \$78.00 (\$39.00) each:	-0-
Multiple Dependent Claim Fee at \$260.00 (\$130.00)	-0-
Total Filing Fee	\$ 690.00

 Assignment Recording Fee \$40.00 -0-

 This application is being filed within the month following the
expiration of the term originally set therefor in the prior application.
This is a petition to request a -month extension of time. A check
covering the cost of the petition is enclosed.

X Oath or Declaration and Power of Attorney

 New signed unsigned

 X Copy from a prior application (37 C.F.R. 1.63(d))

08/31/00 09/652748

Patent Application Transmittal

(only for new nonprovisional applications under 37 C.F.R. 1.53(b))

450117-4278.2

Deletion of Inventors

____ Signed Statement attached deleting inventor(s) named in the prior application (37 C.F.R. 1.63(d)(2) and 1.33(b))

X Incorporation by Reference (for continuation or divisional application)
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

X A Preliminary Amendment is enclosed.
(Claims added by this amendment have been properly numbered consecutively beginning with the number next following the highest numbered original claim in the prior application.)

X Cancel in this application original claims 1-36 and 49-78 of the prior application before calculating the filing fee. (At least one original independent claim must be retained for filing purposes.)

X New formal drawings are enclosed.

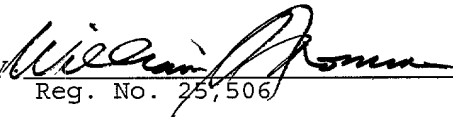
X Certified copy of each foreign priority application on which the claim for priority under 35 U.S.C. 119 is based was filed in prior U.S. application serial no. 08/979,474, filed November 26, 1997. A list of said foreign priority application(s) is provided below. Acknowledgement thereof is requested.

<u>Application No.</u>	<u>Filed</u>	<u>In</u>
96203338.7	27 November 1996	EP
96203341.1	27 November 1996	EP
96203340.3	27 November 1996	EP
96203339.5	27 November 1996	EP
96203334.6	27 November 1996	EP
96203336.1	27 November 1996	EP

Please charge any additional fees required for the filing of this application or credit any overpayment to Deposit Account No. 50-0320.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP
Attorneys for Applicants
WILLIAM S. FROMMER

By 
Reg. No. 25,506

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Division of Serial No. 08/979,474

Applicants : PAUL HODGINS, et al.

Filed : Herewith

For : METHOD AND APPARATUS FOR SERVING DATA

Examiner : Nguyen, P.

Art Unit : 2739

745 Fifth Avenue
New York, New York 10151
Tel. (212) 588-0800

EXPRESS MAIL

Mailing Label Number EL588274972US
Date of Deposit August 31, 2000
I hereby certify that this paper or fee is being
deposited with the United States Postal Service
"Express Mail Post Office to Addressee" Service
under 37 CFR 1.10 on the date indicated above and
is addressed to the Assistant Commissioner for
Patents, Washington, D.C. 20231.

Charles J. Jackson
(Typed or printed name of person
mailing paper or fee)

Charles J. Jackson
(Signature of person mailing paper or fee)

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Box Patent Application
Washington, D.C. 20231

Sir:

Before the issuance of the first Official Action,
please amend the above-identified application as follows:

IN THE SPECIFICATION:

Please amend the specification as follows:

Page 1, line 1, please insert --This is a divisional of U.S.
Patent Application Serial No. 08/979,474, filed November 26,
1997.--

IN THE CLAIMS:

Please cancel claims 1-36 and 49-78


REMARKS

This preliminary amendment makes reference to the
parent application and cancels those claims that are being
prosecuted in the parent. Entry of the above amendatory matter
and early examination on the merits are respectfully requested.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP
Attorneys for Applicants

By


William S. Frommer
Registration No. 25,506
Tel. (212) 588-0800

PATENT
450117-4278

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

TITLE: METHOD AND APPARATUS FOR SERVING DATA

INVENTORS: PAUL HODGINS, GERT JOSEF ELISA COPEJANS, JOERI
APTS, JOHAN DE VOS

William S. Frommer
Registration No. 25,506
FROMMER LAWRENCE & HAUG LLP
745 Fifth Avenue
New York, New York 10151
Tel. (212) 588-0800

METHOD AND APPARATUS FOR SERVING DATA

The present invention relates to servers for data delivery. Traditional servers were designed with the tendency to be actively involved in the physical transmission of data. For applications such as video on demand or karoake on demand, deliverance of a high number of digital video streams in real time are required. The digital video stream typically include video data compressed according to ISO/IEC 11172 or ISO/IEC 13818, which are commonly known as MPEG-1 standard and MPEG-2 standard respectively.

An ATM (Asynchronous Transfer Mode)-base server system with capabilities extending beyond mere data delivery has already been proposed in the European Patent Application No. 95.200819.1.

A streaming engine for ATM communication configured as an ASIC (Application Specific Integrated Circuit) has been proposed in the international application WO 96/08896 published under PCT.

A method for recording and reproducing compressed video data according to the MPEG standard is proposed in European Patent Application EP 0 667 713 A2. In this case, the compressed video data is recorded in the disk in the special form including the scan information so that the specific compressed video data reproducing apparatus can achieve VCR functions (e.g. FF, FR).

It is an object of the present invention to improve upon the above mentioned prior art and/or to provide a server for future applications.

The present invention provides in a first aspect a method for translating a VPI/VCI of an ATM cell into an internal ID comprising the steps of:

distributing VPI/VCI entries into sections in a table according to a portion of each VPI/VCI entry;
receiving an ATM cell;

5

10

15

20

This apparatus according to the present invention provides for management of running applications that interact with a large number of clients and management modules distributed over a system, as well as management of the data that are delivered. The server according to the present invention provides time or processing power to run higher level management tasks, as the host is less actively involved in physical transmission of data. The hardware according to the present invention is able to deliver data in real time under different performance requirements and is well suited for such real time delivery. The streaming engine according to the present invention is able to support simultaneous communications with many clients and to facilitate the video streaming task. The server according to the present invention, also provides for interoperability, such as to serve data to any type of

client. The content to be delivered, can be stored in a versatile form (i.e. raw or non formatted form) in the server according to the present invention.

The present invention provides in a third aspect a method for streaming data from a storage device comprising the steps of:

providing write addresses for a burst data to a buffer, at least a portion of the write addresses being non-contiguous;

transferring the burst data from the storage device to the buffer via a bus supporting communication between a host, the storage device and a streaming device;

writing the burst data in the buffer according to the write addresses; and

reading data from the buffer in a linear fashion.

Preferred embodiment of the method according to the present invention are described in the dependent sub-claims.

Further the present invention provides a streaming device for streaming a data from a storage device comprising:

means for receiving a burst data from the storage device via a bus supporting communication between a host, the storage device and the streaming device;

means for providing write addresses for the burst data, at least a portion of the write addresses being non-contiguous; and

a buffer for storing the burst data according to the write addresses and outputting data therefrom in a linear fashion.

The present invention provides in a fourth aspect a method for delivering data comprising the steps of:

loading at least a pair of an address and a command from a host;

storing the data in a buffer;

reading the data from the buffer according to a read pointer;

executing the command if a match between the address and an address specified by the read pointer is
5 detected; and

delivering the data read from the buffer after the execution of the command.

Further the present invention provides a device for delivering data comprising:

10 a command block for storing at least a pair of an address and a command loaded from a host and detecting a match between the address and an address specified by a read pointer of a buffer buffering the data;

means for executing the command in cooperation
15 with the command block when the match is detected; and

means for delivering the data read from the buffer after the execution of the command.

The present invention provides in a fifth aspect a method for delivering data comprising the steps
20 of:

receiving data from a network;

detecting at least a preset bit pattern in the data when the received data is transmitted to a storage device;

25 adding location information corresponding to a location of the preset bit pattern in the data to a list when the preset bit pattern is detected;

storing the data in the storage device; and

controlling a delivery of the data from the
30 storage device to the network according to the location information in the list.

Preferred embodiment of this method are described in the dependent subclaims.

Further the present invention provides an
35 apparatus for delivering data comprising:

receiving means for receiving data from a network;

a pattern detector for detecting at least a preset bit pattern in the data when the data is transmitted from the receiving means to a storage device storing the data;

5 a list for storing location information corresponding to a location of the preset bit pattern in the data when the preset bit pattern is detected by the pattern detector; and

means for controlling a delivery of the data
10 from the storage device to the network according to the location information in the list.

The present invention provides in a sixth aspect a traffic shaping method comprising the steps of:

classifying one or more first streams into one
15 or more classes, each class including one or more streams having the same bit rate characteristics;

setting a set of parameters to control the bit rate for each class; and

executing a rate pacing of each class according
20 to the set of parameters.

Preferred embodiment of this method are described in the dependent subclaims.

Further the present invention provides a traffic shaper comprising:

25 means for classifying one or more first streams into one or more classes, each class including one or more streams having the same bit rate characteristics;

storage means for storing a set of parameters to control the bit rate for each class; and

30 means for executing a rate pacing of each class according to the set of parameters in the storage means.

Further advantages, features and details of the present invention will become clear when reading the following description, in which reference is made to the
35 annexed drawings, in which:

figure 1 shows a general system architecture of an interactive communication system;

figure 2 shows a detail block diagram of an embodiment of the apparatus according to the present invention;

figure 3 shows a block diagram of the Tx
5 address translator of figure 2;

figure 4 shows an example of the use of the address translator of figure 3;

figure 5A, 5B, 5C show respective examples of address translation for TCP IP packetisation;

10 figure 6 shows an example of use of the Tx rate block of figure 2;

figure 7 shows the behaviour of a bit rate achieved by the traffic shaper of figure 2;

figure 8 shows a diagram for explaining the
15 sending of stream within one cell period;

figure 9 shows the submission of cells for different classes;

figure 10 is a block diagram of an architecture for the traffic shaper of figure 2;

20 figure 11 shows a block diagram of the command block of figure 2;

figure 12 is a diagram for explaining the operation of the byte swapper of figure 2;

figure 13 shows a format of one ATM cell used
25 in UNI;

figure 14 shows a block diagram of the VPI/VCI translator of figure 2;

figure 15 shows a block diagram of the pattern detector of figure 2; and

30 figure 16 shows an example of an address translator of figure 2.

Figure 1 shows a general system architecture of a preferred embodiment of an interactive communication system. This is a broad-band system that supports
35 virtually any kind of interactive multi-media application. Particular attention is paid to real time multimedia delivery mode applications.

A server 10 functions as VOD (Video On Demand) server, KOD (Karaoke On Demand) server, and/or Internet server, etc. and communicates with STBs (Set Top Box) 18 as clients over a public network 16. The server 10 consists of a local ATM switch 14 and several SMUs (Storage Medium Unit) 12 that are interconnected thorough the local ATM switch 14. The main purposes of the local ATM switch 14 are to route data between the SMUs 12 (for instance, to duplicate a movie compressed according to the MPEG standard from one SMU to another), create a ATM-based LAN inside the server 10, and interface to the public network 16. Each SMU 12 communicates with the local ATM switch 14 at high speed, with current technology at e.g. a maximum of 622 Mbps. The public network 16 is optional and the server 10 may directly communicates with the STBs 18.

Figure 2 shows a detail block diagram of the SMU 12. The SMU 12 has storage devices 20, a host 28 and a streaming engine 36 as major units. These units are interconnected via a PCI (Peripheral Component Interconnect) bus 24. A host CPU 30 and a host memory 32 in the host 28 are connected via MIPS bus 34 in a conventional configuration. In this embodiment the MIPS bus 34 is connected to the PCI bus 24 thorough a PCI bridge 26. The host 28 is primarily intended for running applications like VOD, KOD, internet server that interact with clients or STBs.

The storage devices 20 contains one or more strings of hard disks. These hard disks are connected via SCSI or Fibre Channel and store real time sensitive data like MPEG-2 encoded video streams and the contents of data packets like the body of TCP/IP (Transmission Control Protocol/Internet Protocol) packets without the headers.

The streaming engine 36 is preferably configured as a single ASIC (Application Specific Integrated Circuit). The streaming engine 36 streams the real time sensitive data and the data packets. The

streaming engine 36 has a transmission path 50 and a reception path 80 as major parts and a PCI interface 38 and an interface 40. The transmission path 50 handles the outgoing data stream from the storage devices 20 and the host 28 to the local ATM switch 14. The reception path 80 handles the incoming data stream from the local ATM switch 14 to the storage devices 20 and the host 28. The high speed connections and the independence of the transmission path and reception path allow for 622Mbps simultaneously in both directions.

The PCI interface 38 interfaces the PCI bus 24 with the transmission path 50 and the reception path 80. The PCI interface 38 transfers the outgoing data stream from the PCI bus 24 to a PCI FIFO 52 in the transmission path 50 and transfers the incoming data stream from a PCI FIFO 98 in the reception path to the PCI bus 24.

The interface 40 interfaces the transmission path 50 and the reception path 80 with an external physical layer device (not shown) connected to the local ATM switch 14. The interface 40 can include two types of ATM interfaces according to the UTOPIA (Universal Test and Operation PHY Interface for ATM) level 2 standard. One is UTOPIA interface in 8 bit wide data path mode and the other is UTOPIA interface in 16 bit wide data path mode.

The transmission path 50 consist of several functional blocks which act together to perform high speed transmission.

The first block in the transmission path 50 is the Tx address translator 54, which places the outgoing data stream from the PCI FIFO 52 into host-specified memory locations of a stream buffer 44 allocated in an external RAM 42. This allows for controlled "scattering" of data into non-contiguous memory, which is useful for an operation which to some extent resembles so-called RAID (Redundant Array of Inexpensive Disks) operation which ensures integrity of data streams and TCP/IP packetisation.

The TCP/IP checksum block 56 provides hardware support for calculating TCP/IP checksums. Its function is to calculate and maintain a partial checksum for each packet until all data has been transferred. The TCP/IP
 5 checksum block 56 works together with the Tx address translator 54 to create TCP/IP packets directly in the stream buffer 44. The TCP/IP header and payload of the packets are placed in the stream buffer 44 separately, passing through the checksum block 56 which keeps a
 10 partial checksum. As soon as all data is in the stream buffer 44, the checksum value is placed in the correct position of TCP/IP header, the packet is ready for transmission.

The RAM interface 58 forms an interface between
 15 the external RAM 42 and the transmission path 50. The external RAM 42 may comprise dual ported SDRAM (Synchronous Dynamic RAM). This external RAM 42 includes several stream buffers 44 to decouple the bursty data traffic from the disks of the storage devices 20 and
 20 provides the required constant bit rate data streams to the ATM-network 16. Each stream buffer handles one outgoing data stream. In the contrast to the incoming direction, since the data flow characteristics in the outgoing direction are fully predictable (controllable),
 25 the buffer requirements can be estimated beforehand. Therefore the stream buffers 44 are statically allocated in the external RAM 42.

The Tx RAID or SDI (Stream Data Integrity) block 60 provides support for data redundancy. The Tx
 30 address translator 54 places the data as needed in stream buffer 44. Then, as data is output from the stream buffer 44, the Tx RAID block 60 corrects error data in the event that one of the disks in the storage devices 20 breaks down.

35 The traffic shaper 62 controls the streaming of the outgoing data from the stream buffers 44 to the ATM network 16. It is designed for very accurate rate pacing and low CDV (Cell Delay Variation). The traffic shaper 62

consists of two main sections. One section handles high priority data such as video traffic, and the other section handles general data traffic of low priority.

The command block 66 is intended to off-load the host especially 28 of real-time sensitive jobs. It performs actions triggered by the transmission of the content of exact known locations in the outgoing data stream.

The segmentation block 70 segments the outgoing data stream provided from the stream buffer 44 into AAL-5 PDUs (ATM Adaptation Layer -5 Protocol Data Units), and maps the AAL-5 PDUs into ATM cells. In case the outgoing data stream is MPEG-2 SPTS (Single Program Transport Stream), the segmentation block 70 is able to segment two TS packets in the MPEG-2 SPTS to one AAL-5 PDU, unless there are less than two TS packets left in the MPEG-2 SPTS, in the latter case the AAL-5 PDU maps into eight ATM cells. In the general case, the AAL-5 segmentation is controlled by the PDU size which is programmable per stream.

The reception path 80 has several blocks corresponding to the reverse operation of the blocks of transmission path 50.

A VPI/VCI (Virtual Path Identifier/Virtual Channel Identifier) filtering block 84 performs fast and efficient VPI/VCI filtering of the incoming ATM cells. This is done by a combined hash and linear search functions over the entries in a VPI/VCI table.

A reassembly block 86 basically performs the inverse functions of the segmentation block 70. The reassembly block 86 reconstructs the AAL-5 PDUs using payload of the ATM cells, then maps the AAL-5 PDUs into the upper layer data (e.g., MPEG-2 SPTS, TCP/IP Packets).

A TCP checksum verification block 88 verifies the TCP checksum in the TCP header if the incoming data stream is transmitted via TCP.

A pattern detector 92 allows a limited number of bit patterns to be detected in an incoming data

stream. A list is created, indicating exactly where the specified bit patterns occur in the stream. This supports certain processing tasks that can be performed on-the-fly, whereas they would otherwise have to be done with post-processing.

A Rx RAID or SDI block 90 adds redundancy to the incoming data stream. If a sequence of N words is written to a buffer (not shown), the parity over these N words is written next. This function can be turned on/off. If the incoming data stream will be stored in the storage device 20 and transmitted later as TCP/IP packets via the transmission path 50, the function is turned off.

A RAM interface 94 is an interface between the reception path 80 and an external RAM 46. The external RAM 46 may comprise dual ported SDRAM. The external RAM 46 is used as several stream buffers 48 storing incoming data streams. Each stream buffer 48 handles one incoming data stream. Incoming data streams can have unpredictable properties. For instance, some of data packets can be very bursty. This means the required buffer capacity varies from stream to stream and from time to time. Therefore, In the external RAM 46, a dynamic buffer allocation is preferred.

A Rx address translator 96 provides appropriate read addresses to the stream buffer 48.

The details of the major blocks in the streaming engine 36 are described below.

Tx Address Translator

The outgoing data stream is provided from the storage device 20 to the streaming engine 36 in burst transmission over the PCI bus 24. The purpose of the Tx address translator 54 is to scatter one contiguous DMA burst in appropriate areas of the stream buffer 44.

Figure 3 shows a block diagram of the Tx address translator 54. Before one contiguous DMA burst from the storage device 20 arrives, the correct starting address is written to a register 102 via a storage device

controller 22. The content of the register 102 is used as write address for the stream buffer 44. A counter 106 counts the number of bits of the outgoing data stream from the PCI FIFO 52. Each time a data word consisting of 32 bits passes the counter 106, it inform a increment controller 104 that a word is transferred to the stream buffer 44. With each new word, the increment controller 104 increments the content of the register 102 with ADDRESS_INCREMENT, which is a programmable value. In case of the outgoing data stream being RAID processed data, the value of ADDRESS_INCREMENT is basically set according to the number of disks used for RAID system. In case of the outgoing data stream being payload of a TCP/IP packet, the value of ADDRESS_INCREMENT is basically set according to packetisation parameters.

An address translation when the outgoing data stream is RAID processed data, is described below with reference to Figure 4. In this example, the RAID or SDI system consists of four disks Disk 0, Disk 1, Disk 2 and Disk 3. The Disk 0 contains words 1, 4, 7... to be transmitted to the local ATM switch 14. The Disk 1 also contains words 2, 5, 8... to be transmitted to the local ATM switch 14. The Disk 2 also contains words 3, 6, 9... to be transmitted to the local ATM switch 14. The Disk 3 contains parity words 0, 1, 2... for error correction. Each parity word (e.g., parity 0) has been generated in the Rx RAID block 90 from three words (e.g., words 1, 2 and 3) which constitute so-called stripe unit of RAID together with the parity word.

In the event of failure in one of the disks (e.g., Disk 2), one contiguous DMA burst including parity words is transferred to the Tx address translator 54. For ease of explanation, assume that the size of one contiguous DMA burst is 96 bytes (24 words), although the actual size can be larger than 100k bytes (depending on the speed of the hard- and/or software). In this case, the contiguous DMA burst 120 consists of words 1, 4, 7, 10, 13, 16 from the Disk 0, words 2, 5, 8, 11, 14, 17

from the Disk 1, words 3, 6, 9, 12, 15, 18 from the Disk 2, and parity words 0, 1, 2, 3, 4, 5 from the Disk 3. The Tx address translator 54 generates the following sequence of addresses:

5

178, 182, 186, 190, 194, 198 (data from Disk 0)
 179, 183, 187, 191, 195, 199 (data from Disk 1)
 180, 184, 188, 192, 196, 200 (data from Disk 2)
 181, 185, 189, 193, 197, 204 (data from Disk 3)

10

More specifically, before the contiguous DMA burst 120 arrives at the stream buffer 44, a value 178 is stored in the register 102 as the starting address. Then the word 1 from Disk 0 is written in the address 178 in the stream buffer 44. When the word 1 passes the counter 106, the increment controller 104 increments the value 178 in the register 102 with ADDRESS_INCREMENT of a value 4 corresponding to the number of disks. Then the word 4 from the Disk 0 is written in the address 182 in the stream buffer 44. When the word 4 passes the counter 106, the increment controller 104 increments the value 182 in the register 102 with ADDRESS_INCREMENT of a value 4. Then the word 7 from the Disk 0 is written in the address 186 in the stream buffer 44. Similarly, remaining words 10, 13 and 16 from Disk 0 are written in the addresses 190, 194, 198 which are the number of disks apart in the stream buffer 44.

When the word 16 from Disk 0 passes the counter 106, the increment controller 104 increments the value 198 in the register 102 with ADDRESS_INCREMENT of a value - 19. Then the word 2 from Disk 1 is written in the address 179 in the stream buffer 44. When the word 2 passes the counter 106, the increment controller 104 increments the value 179 in the register 102 with ADDRESS_INCREMENT of a value 4. Then the word 5 from Disk 1 is written in the address 183 in the stream buffer 44. When the word 5 passes the counter 106, the increment controller 104 increments the value 183 in the register

102 with ADDRESS_INCREMENT of a value 4. Then the word 8
 from Disk 1 is written in the address 187 in the stream
 buffer 44. Similarly, remaining words from Disk 1 are
 written in the addresses 191, 195, 199 which are the
 5 number of disks apart in stream buffer 44.

In the same way, words from the Disks 2 and 3
 are written in appropriate addresses in the stream buffer
 44. The words written in the stream buffer 44 are read in
 liner fashion and provided to the Tx RAID block 60 to
 10 correct errors.

When the outgoing data stream from the storage
 device 20 is TCP/IP payload, the address translator 54
 and the TCP checksum calculation block 56 work closely
 together to provide support for TCP/IP packet generation.
 15 The host 28 pre-programs the Tx address translator 54 so
 that data is distributed according to a specified packet
 size. At first the host 28 needs to know all the
 packetisation parameters. Important parameters for this
 operation are TCP payload size, TCP header size, IP
 20 header size and IP payload size. TCP header and IP header
 basically have space for optional data but this is in
 practice not used. Therefore, a simplification can be
 introduced by assuming default sizes for the headers: TCP
 header size is 5 words (20 bytes) and IP header size is 5
 25 words (20 bytes).

The mechanism can be described as follows.

The host 28 itself does a partial checksum
 calculation over the pseudo-header of the TCP/IP header.
 Then it initializes a TCP checksum register 57 in the TCP
 30 checksum block 56 for that TCP/IP packet with this value.
 Space for the stream buffer 44 also will be reserved in
 the external RAM 42 to fit the full TCP packet plus the
 TCP and IP header overhead.

The host 28 will then instruct the increment
 35 controller 104 in the Tx address translator 54 with the
 TCP payload size, TCP header size, IP header size and IP
 payload size. The TCP payload can then be sent as one
 contiguous DMA burst over the PCI bus 24 and placed into

the area in the stream buffer 44 reserved for it by the Tx address translator 54, leaving space for the headers. As it goes from the PCI bus 24 to the stream buffer 44, the checksum calculation block 56 updates the partial

5 checksum in the TCP checksum register 57. Note that with this method the payload, representing usually the bulk of the TCP/IP packets, does not need to be copied first from the storage devices 20 to the host memory 32 for processing it and then to the stream buffer 44. This

10 saves valuable bus bandwidth and overhead for the host CPU 30. After the payload has been written, the header information, prepared by the host 28, is sent to the stream buffer 44 via the address translator 54. As with the payload, the Tx address translator 54 places the

15 header in the previously reserved memory locations.

This sequence can be reversed, whereby the header information is written first and the payload second.

In either case, when both the header and the

20 payload have been written, the TCP checksum will be complete and can be copied to the correct location automatically.

This mechanism can also be used to efficiently support segmenting of a TCP packet into multiple smaller

25 IP packets. In this case, space is reserved for each IP packet. The TCP packet data (header + payload) is segmented into these packets and the header of each IP packet is written by the host 28.

All IP packets will be the same size except for

30 the last block, which is likely to have a different size than the others. The address translator 54 takes this in to account. After the complete TCP/IP packet(s) has been formed, it is ready for transmission.

Figure 5A, 5B and 5C shows an example of

35 address translation for TCP/IP packetisation. In this case, before the TCP/IP payload sent as one contiguous DMA burst 130 arrives at the stream buffer 44, a value 310 is stored in the register 102 as the starting write

address, then the first word of the first data is written in the address 310 in the stream buffer 44. When the first word of the first data passes the counter 106, the increment controller 104 increments the value 310 in the register 102 with ADDRESS_INCREMENT of value 1. Then the second word of the first data is written in the address 311 in the stream buffer 44. When the second word of the first data passes the counter 106, the increment controller 104 increments the value 311 in the register 102 with ADDRESS_INCREMENT of value 1. Then the third word of the first data is written in the address 312 in the stream buffer 44. The increment with ADDRESS_INCREMENT of value 1 is repeated a number of times corresponding to the IP payload size. Thus the first data of the TCP/IP payload is written in an appropriate area.

Then the increment controller 104 increments the content in the register 102 with ADDRESS_INCREMENT of a value corresponding to IP header size. Then the writing of second data starts from the address according to the content of the register 102. Thus the address translator 54 generates write addresses for the payload so that the space for the headers are left. The last data is likely to have a different size than the others. The size of the last data is calculated in the increment controller 104 by the following expression:

The last data size = TCP payload size mod IP payload size

Therefore the number of increment is controlled taking the last data size into account. In this way, the payload sent as one contiguous DMA burst is scattered in the shaded areas in the stream buffer 44 shown as Figure 5A.

Next, When the TCP header 132 is sent as one contiguous burst over the PCI bus 24, the address translator 54 generates write addresses corresponding to the previously reserved memory locations for the TCP header in the stream buffer 44.

More specifically, before the TCP header sent as one contiguous burst 132 arrives at the stream buffer 44, a value 305 is set in the register 102 as the starting write address, whereafter the first word of the TCP header is written in the address 305 in the stream buffer 44. When the first word of the TCP header passes the counter 106, the increment controller 104 increments the value 305 in the register 102 with ADDRESS_INCREMENT of value 1. Then the second word of the TCP header is written in the address 306 in the stream buffer 44. When the second word of the TCP header passes the counter 106, the increment controller 104 increments the value 306 in the register 102 with ADDRESS_INCREMENT of value 1. Then the third word of the TCP header is written in the address 307 in the stream buffer 44. The increment with ADDRESS_INCREMENT of value 1 is repeated a number of times corresponding to the TCP header size. Thus the TCP header is written in the shaded area in the stream buffer 44 shown as Figure 5B.

Next, When the IP headers 134 are sent as one contiguous burst over the PCI bus 24, the address translator 54 generates write addresses corresponding to the previously reserved memory locations for the IP headers in the stream buffer 44.

More specifically, before the IP headers sent as one contiguous burst 134 arrives at the stream buffer 44, a value 300 is set in the register 102 as the starting write address, whereafter the first word of the first IP header is written in the address 300 in the stream buffer 44. When the first word of the first IP header passes the counter 106, the increment controller 104 increments the value 300 in the register 102 with ADDRESS_INCREMENT of value 1. Then the second word of the first IP header is written in the address 301 in the stream buffer 44. When the second word of the first IP header passes the counter 106, the increment controller 104 increments the value 301 in the register 102 with ADDRESS_INCREMENT of value 1. Then the third word of the

first IP header is written in the address 302 in the stream buffer 44. The increment with ADDRESS_INCREMENT of value 1 is repeated a number of times corresponding to the IP header size.

- 5 Then the increment controller 104 increments the content in the register 102 with ADDRESS_INCREMENT of a value corresponding to TCP header size + IP payload size. Then the writing of second IP header starts from the address according to the content of the register 102.
- 10 Thus the IP headers are written in the shaded areas in the stream buffer 44 shown as Figure 5C.

Next, the TCP checksum completed by the TCP checksum block 56 is copied to the correct location.

- 15 In this way, the TCP/IP packetisation is completed and can be read from the stream buffer 44 in linear fashion.

- In the above embodiment, TCP/IP packetisation is mentioned. However it is possible to use UDP (User Datagram Protocol) instead of TCP. In this case, the
- 20 default size of UDP header is 2 words (8 bytes).

- In addition, in the above embodiment, the TCP header and the IP headers are sent as different bursts from the host 28 to the Tx address translator 54. However it is possible to send the TCP header and the IP headers
- 25 together as one contiguous burst from the host 28 to the Tx address translator 54.

Tx RAID or SDI block

- In the sequence of words in the stream buffer
- 30 44, parity words may be inserted. This redundancy provides a means for correcting errors. The Tx RAID or SDI block 60 takes in a sequence of N+1 words of which the last word is the parity over the N first words. In case it is indicated by hardware and/or software, that
- 35 word M is corrupt, e.g., because of a disk failure, the parity word is retrieved from the storage device 20 and used to reconstruct the word M.

For example, in case of Figure 4, the words 3, 6, 9, 12, 15, 18 from the failure Disk 2 in the input data 142 include error shown as Figure 6. The Tx RAID block 60 reconstruct the word 3 using the words 1, 2 and the parity word 0. The Tx RAID block 60 reconstruct the word 6 using the words 4, 5 and the parity word 1. Similarly, the words 9, 12, 15, 18 are reconstructed by the Tx RAID block 60. Thus the Tx RAID block 60 performs error correction and outputs the sequence 142 of the words 1, 2, 3, 4 ... without errors.

The RAID function can be turned on/off by the command block 66.

15 Traffic Shaper

The traffic shaper 62 consists of two main sections one section handles high priority data such as video traffic, and a low priority section handles general data traffic.

The high priority section is organized into several traffic classes, in which a class is a group of one or more streams having the same bit rate characteristic. For example, all streams of a CBR (Constant Bit Rate) at 2 Mbps belong to the same class. A class of VBR (Variable Bit Rate) type typically contains only one stream, because it is unlikely that two VBR streams have identical bandwidth patterns at all times. Each class has a single set of transmission parameters for controlling the bit rate, providing for low CDV (Cell Delay Variation) and accurate rate pacing. The number of classes is programmable but limited to maximum 128.

Each class has two main transmission parameters, an ideal scheduled time (TS) and an increment (Δ) for the TS. The basic mechanism is that when TS becomes equal to or less than a reference clock, a stream pointer is put into the transmit queue. At the same time the value TS is incremented with the value Δ . The transmit queue is a first in first out queue that will

submit the stream indicated by the stream pointer the ATM fifo 72 as soon as possible.

In the high priority section a high accuracy bit rate and low CDV are achieved following mechanisms.

5 Due to the finite resolution of the reference clock, having a single Δ value usually does not give the desired accuracy. To achieve the desired accuracy, two Δ values are used alternatively that are just one counter value apart. These two values result in a rate that is
10 slightly above and below the required bit rate. Using each Δ value for different numbers of cells compensates for the limited clock resolution and can provide arbitrary accuracy. Δ_H and Δ_L (where $\Delta_L = \Delta_H + 1$) represent the two different increment values. The N_H and N_L parameters
15 represent the number of cells for which the corresponding increment value are alternatively valid. By means of this mechanism, the stream is modulated whereby the average bit rate approaches the required bit rate within the desired accuracy. Figure 7 shows a behavior of a bit rate
20 achieved by this mechanism. In Figure 7, N_H cells are sent at Δ_H and N_L cells are sent at Δ_L . This sequence is repeated cyclically. Thus the average bit rate shown by a dotted line is maintained as a long term bit rate.

Low CDV is achieved by reducing the collisions
25 in scheduling times of cells from different streams. A major cause of collisions in many existing traffic shaping mechanisms is that streams of the same bit rate are scheduled at the same time. This is particularly a problem when there is a large number of streams and a low
30 number of independent bit rates. This problem is addressed in the preferred embodiment by evenly spacing the cells of streams belonging to the same class. In other words, if the increment for one stream should be Δ , the increment for the class is Δ/n , where n is the number
35 of streams in a class. Every time a class is to be serviced, the data is taken from successive streams. For example, If cells of the stream 0 belonging to Class 0 should be incremented by Δ , each cell of the streams

(i.e. stream 0 - stream n-1) belonging to the same class 0 is sent with the space of Δ/n shown as Figure 8.

By the combination of the above two mechanisms, a high accuracy bit rate and low CDV are achieved. If the transmit queue does not get blocked, the cells are submitted shown as Figure 9.

The high priority section also handles VBR traffic. The traffic shaper 62 supports smooth update of the transmission parameters. This update can be done by the host 28 but also by the command block 66. The command block 66 is programmed by the host 28, and its actions are triggered when an exact location is transmitted from the stream buffer 44. One such action is to replace the transmission parameters for a specified stream in the traffic shaper 62. As soon as the data just before a change in bit rate are sent, the command block 66 updates the parameters. Once it is set-up, this process is autonomous and does not require interaction of the host CPU 30 anymore. As a consequence, the host 28 does not need to interact exactly at the moment interaction would be required. In this way the real-time character of the stream is maintained and the host load kept to a minimum.

The low priority section is organized into e.g. fixed 32 traffic classes, in which a class is a group of one or more streams having the same PCR (Peak Cell Rate). In terms of the general data traffic, the real-time constraints are much less significant. The main objective of the traffic shaping of the low priority section is to limit the PCR in order to avoid network policing. The traffic shaping of the data packets is implemented by a mechanism using an ideal scheduled time (TS) and an increment (Δ) for the TS, which is similar to the basic traffic shaping mechanism in the high priority section. However, scheduling of data packets gets a lower priority than real time traffic. Only if the transmit queue of the high priority section is empty, a stream of the data packets can be submitted to the ATM FIFO 72.

The mechanism is implemented with an architecture shown as Figure 10. The traffic shaper 62 consists of the high priority section 200 and the low priority section 202 as mentioned above.

5 In the high priority section 200, a memory 203 stores a set of new transmission parameters for each class provided from the host 28. Each set of the new transmission parameters consists of TS_i , Δ_{Hi} , Δ_{Li} , N_{Hi} , N_{Li} and Pt_i (where $0 \leq i \leq 127$). In this embodiment Pt_i contains one or more stream pointers which indicate one or more streams attached to the class i . A memory 206 stores current transmission parameters. When a command is instructed by the host 28 or the command block 66, an update logic 204 is triggered by the command, whereby the current transmission parameters in the memory 206 are updated with the new transmission parameters in the memory 203. A register 212 stores a parameter $Nr_Classes$ indicating the number of class from the host 28 at receipt thereof. A traffic logic 208 checks for each of classes from 0 to $Nr_Classes-1$ whether TS_i is equal to or less than the current time indicated by a reference clock 210. If so, the stream pointer of the first stream attached to this class i is inserted to a high priority transmit queue 216 and TS_i in the memory 206 is incremented with Δ_{Hi} or Δ_{Li} of this class i by the traffic logic 208. The Δ_{Hi} and Δ_{Li} are alternated according to the N_{Hi} and N_{Li} . Then the segmentation block 70 receives the stream pointer from the high priority transmit queue 216 and puts a ATM cell belonging to the stream indicated by the stream pointer into ATM FIFO 72.

 In the low priority section 202, a memory 218 stores a set of transmission parameters for each class provided from the host 28. In this embodiment each set of the transmission parameters consists of TS_j , Δ_j and Pt_j (where $0 \leq j \leq 31$). Pt_j contains one or more stream pointers which indicate one or more streams attached to the class j . A traffic logic 220 checks each of classes from 0 to 31 if TS_j is equal to or less than the current

time indicated by the reference clock 210 and monitors where the high priority transmit queue 216 is empty. If so, the stream pointer of the first stream attached to this class j is inserted to a low priority transmit queue 5 222 and TS_j in the memory 218 is incremented with Δ_j of this class j by the traffic logic 220. Then the segmentation block 70 receives the stream pointer from the low priority transmit queue 222 and puts a ATM cell belonging to the stream indicated by the stream pointer 10 into ATM FIFO 72.

In the above embodiment, a traffic shaping mechanism being similar to the mechanism of high priority section 200 is applied to the low priority section 202. However conventional leaky bucket mechanism may be 15 applied to the traffic shaping mechanism of the low priority section 202.

Command Block

Real time data delivery may sometimes involve 20 actions occurring at specific locations in the outgoing data stream. These actions must be immediate in order to maintain the integrity of the stream. Due to the many responsibilities of the host 28, timely interaction cannot always guaranteed. In the preferred embodiment, it 25 is the responsibility of the command block 66 to perform these interactions. In principle, the host 28 knows exactly where in the outgoing data stream the streaming parameters need to be adjusted. Since each stream buffer 44 is allocated statically as mentioned above, it is 30 possible to express a location, where the actions should be taken, in read pointer of the stream buffer 44. The host 28 loads a list of the command block 66 with a number of instructions at the appropriate moment in time. The appropriate time is the time between the loading of 35 the outgoing data stream to the stream buffer 44 and the time that the outgoing data stream is sent out from the stream buffer 44. The command block 66 scans the read pointer of the stream buffer 44. If a match with a

specified address is found, a command that is linked to that address will be executed and the command will be purged from the command block 66.

The command block 66 triggers on the address of the data leaving the stream buffer 44. When reading a stream buffer 44, the read pointer is gradually incremented with a wrap-around. Each stream has a linked list that contain the (address, command) pairs to be stored according to the address. An address is a pair (L, M) indicating the address in the file and is independent from the physical address. L is the number of blocks, with a block size equal to the size of the stream buffer 44. M is sequence number in the last block. Each stream maintains a WAC (Wrap-around counter) that counts the number of times the read pointer has been wrapped around. An address match is found if

$$L=WAC \quad \text{and} \quad M=\text{Read Pointer} - \text{Buffer Offset}$$

This mechanism is implemented as follows.

Figure 11 shows a block diagram of the command block 66.

The command block 66 consists of several command generators 300. Each command generator 300 handles the commands for each outgoing data stream. The host 28 loads a list of commands in the command register 316 in each command generator 300 at the appropriate moment in time.

In a command generator 300, a register 302 stores the Buffer Offset. A comparator 304 compares the Buffer Offset in the register 302 with the read pointer of the stream buffer 44. When a wrap-around occurs, the read pointer takes the Buffer Offset. Therefore when the match is detected by the comparator 304, the WAC (Wrap-Around Counter) 306 is incremented. The comparator 308 compares the count of the WAC 306 with current L provided from the command register 316. A comparator 310 compares current M provided from the command register 316 with the read pointer - Buffer Offset. When the matches are detected by the comparator 308 and the comparator 310, the AND gate 312 dequeues a current command stored by a queue 314. Each time a current command corresponding

to a current address (L, M) is output from the queue 314, a command corresponding to a next address is queued in the queue 314 from the command register 316. Thus each command generator 300 instructs commands according to a read pointer of the stream buffer 44.

The commands to be instructed from the command block 66 are:

Change bit rate: This command will allow to change a stream bandwidth. When this command is instructed, the traffic shaper 62 detach a stream from its current class, updates the Δ values for the current class, attach the stream to a new class and update the linked Δ values of the new class. Thus the bit rate of individual streams is changed at specific stream locations. This is useful for MPEG bit stream of VBR (variable bit rate), for example.

Insert RCI: This command allows to insert an RCI (Rate Change Indicator) at specific location in the stream. The RCI is able to notify the distant terminal (e.g., STB 18) the rate changes at that moment and aids clock recovery for MPEG decoders. The detail of the RCI is described as "data rate data" in the European Patent Application EP 0 712 250 A2. When this command is instructed, the RCI generator 68 generates the RCI and the segmentation block 70 terminates the current segmentation and a separate AAL-5 PDU (one ATM cell) for the RCI is generated. This is useful for MPEG bit stream of VBR.

Enable RAID: This command set the appropriate parameters in the Tx RAID block 60 for the error correction.

Disable RAID: This function is the inverse of the above Enable RAID.

35

Perform Byte Swap: This command allows to cope with little endian / big endian problems between the server 10 and the STB 18. When this command is instructed, the byte

swapper 64 reorders the bytes within a word 350 in the outgoing stream in an order of a word 352 shown as Figure 12.

- 5 Enable different PDU-Size: TCP can require segmentation. One TCP packet is to be divided in different IP-packets. The last IP packet requires usually a different AAL-5 PDU size than the previous one. When this command is instructed, the segmentation block 70 change the AAL-5
10 PDU size.

Interrupt CPU: This is the most general function. It requests the host CPU 30 interaction upon detection of a certain location in the stream.

15

VPI/VCI Filtering Block

- Figure 13 shows a format of one ATM cell used in UNI (User Network Interface). One ATM cell consists of 53 bytes. First 5 bytes constitute a ATM header and the
20 remaining 48 bytes carry payload. The first 4 bits in the ATM header is called GFC (Generic Flow Control). The following 24 bits in the ATM header is called VPI/VCI. Actually, the VPI/VCI consists of VPI of 8 bits and VCI of 16 bits. The following 3 bits in the ATM header is
25 called PT (Payload Type). The following 1 bit in the ATM header is called CLP (Cell Loss Priority). The last 8 bits in the ATM header is called HEC (Header Error Control). The VPI/VCI filtering block 84 receives such ATM cells from ATM FIFO 82.

- 30 The VPI/VCI filtering block 84 determines whether a VPI/VCI of the received ATM cell is an element of the set of VPI/VCIs that should be accepted, determines to which stream the ATM cell belongs, and filters OAM (Operation, Administration and Maintenance)
35 F5 cells. To achieve this filtering process, a VPI/VCI translation from a VPI/VCI to an internal stream ID is performed in a VPI/VCI translator 85 in the VPI/VCI filtering block 84.

The object of the VPI/VCI translation mechanism is to allow as wide a range of legal VPI/VCIs as possible, while at the same time facilitating fast translation. Preferably, all VPI/VCIs should be

5 admissable. The VPI/VCI translation can be done using conventional binary search techniques. However, due to time constraints, the largest acceptable search is of the order of a binary search of 512 entries. On the other hand, the maximum number of active VPI/VCIs should be

10 greater than 512 to support simultaneous communications with a large number of clients.

In order to meet the object, the VPI/VCI table is divided up into sections of 512 entries. Each entry indicates a relation between a VPI/VCI and an internal

15 stream ID is entered into a certain section depending on a distribution mechanism and within each section the entries are ordered.

Upon reception of a ATM cell, once the correct section has been found, a binary search can be performed

20 over that section to find the correct entry. Therefore the distribution mechanism to distribute the VPI/VCIs must allow immediate indexing into a section according to the VPI/VCI. Moreover, to allow for efficient use of the VPI/VCI table, the mechanism must allow for a wide

25 distribution of the VPI/VCIs. In other words, the mechanism must distribute the entries as randomly as possible over the entire VPI/VCI table. If a VPI/VCI maps into a section of the VPI/VCI table that is already full, it must be rejected even though there may be space in

30 other sections.

One distribution mechanism that fits to the requirements is to simply use the lower X (where X is integer; e.g., 3) bits of the VCI as hash key to index into the VPI/VCI table. It is reasonable that when there

35 are a large number of active VP/VCs the lower bits will be the most random of the 24 bits VPI/VCI field and allow for an even distribution.

Using this type of mechanism, the requirements of fast look up and no illegal or inadmissible VPI/VCIs are met. The mechanism is implemented as follows.

Figure 14 shows a block diagram of the VPI/VCI translator 85. When a new VP/VC become active, a new entry indicating a VPI/VCI of that new VP/VC and an internal stream ID corresponding to that VPI/VCI is entered into a section according to the lower 3 bits of the VCI (i.e., bits 7, 6, 5 of 4-th byte in Figure 13) via a hash function 400. More specifically, if the lower 3 bits of the VCI is 000, the entry is stored in the section 1 in the VPI/VCI table 402. If the lower 3 bits of the VCI is 001, the entry is stored in the section 2 in the VPI/VCI table 402. If the lower 3 bits of the VCI is 010, the entry is stored in the section 3 in the VPI/VCI table 402. Similarly, all new entries are stored in appropriate sections according to the lower 3 bits of the VCI. Thus, the VPI/VCI table 402 of e.g. 4096 entries is divided up into 8 sections (section 1 - 8) of e.g. 512 entries. Within each section the entries are reordered in an ascending or descending order to implement a binary search.

Upon reception of an ATM cell, the VPI/VCI of the received ATM cell is provided to a search engine 420 and the hash function 400. The hash function 400 provides a section index based on the lower 3 bits of the VPI/VCI to the search engine 420. Then a binary search is performed by the search engine 420 over a section corresponding to the section index to find the correct entry. For example, if the lower 3 bits of the VCI of the received ATM cell is 010, the hash function 400 provides 3 as the section index to the search engine 420. Then, the search engine 420 performs a binary search over the section 3 to find a correct entry and outputs a internal stream ID of the found entry. If the lower 3 bits of the VCI of the received ATM cell is 111, the hash function 400 provides 8 as the section index to the search engine 420. Then, the search engine 420 performs a binary search

over the section 8 to find a correct entry and outputs a internal stream ID of the found entry. The output internal stream ID is used for the filtering process.

In the above embodiment, the lower 3 bits of the VPI/VCI field is simply used as a section index. However, a more complex hash function may be used over the VPI/VCI field to generate a section index.

In the above embodiment, when a new VP/VC becomes active, the new entry is entered to an appropriate section via the hash function 400. However, it is possible to create a new VPI/VCI table including the new entry in the host 28 having a hash function of the same mechanism as the hash function 400, transfer the new VPI/VCI table to the VPI/VCI translator 85 and update the VPI/VCI table 402 with the new VPI/VCI table.

Pattern Detector

The host 28 knows what kind of data incoming in over a specific VC. The host 28 instruct the pattern detector 92, per VC, which pattern is to be scanned for. The purpose of the pattern detector 92 is to detect a preset bit pattern in the incoming data stream. Each time a match is detected, the pattern detector 92 informs the host 28 the "data detected" state. When the host 28 receives the information of the detection, it adds the address at which it occurs to a list in the host memory 32. As the detection itself is done automatically, the host 28 can perform other jobs in the mean time. The host 28 only needs to be interrupted in case the pre-set bit pattern is detected and the action can be taken.

Figure 15 shows a block diagram of the pattern detector 92. Before the incoming data stream is transmitted through the reception path 80, the host 28 instructs the pattern detect controller 506, per VC, which pattern is to be scanned for. The pattern detect controller 506 can set 4 pre-programmed bit pattern of 32 bits wide in register 504 for each stream. The alignment circuit 500 performs byte alignment of incoming data

stream. The matching circuit 502 performs byte aligned matching against 4 pre-programmed bit patterns per stream. Each time the match is detected, the matching circuit 502 informs the controller 506 of the detection

5 An example of the purpose of pattern detector
92 is to find locations of I-picture in video bit stream
compressed according to the MPEG standard. In the MPEG
bit stream, a picture immediately following GOP header is
always I-picture. Therefore, It is possible to find a
10 location of I-picture by detecting group_start_code (32
bits) identifying the beginning of GOP header and
picture_start_code (32 bits) identifying the beginning of
picture header.

For instance, when a MPEG bit stream of a movie is transferred from another SMU 12 in order to duplicate the movie, group_start_code and picture_start_code are set in the register 504 as pre-set bit patterns. The pattern detector 92 detects group_start_code and picture_start_code in the received MPEG bit stream. Each time picture_start_code is detected immediately after the detection of group_start_code in the matching circuit 502, the pattern detect controller 506 informs the detection state to the host CPU 30. The host CPU 30 adds an address of storage device 20 in which the I-picture is stored to a list in the host memory 32. Thus the list indicting locations of I-picture is constructed during the MPEG bit stream flows in the reception path 80.

The list is used for VCR-operation when the stored MPEG bit stream is transferred to the STB 18. If the STB 18 requests VCR operation (e.g., FF, FR), the host 28 refers this list and instructs the storage device controller 22 to access and retrieve the I-pictures.

Using this feature, the data stored in the storage device 20 is "raw" or not formatted for a specific application. This increases the "application independence" and interoperability of server system 10 (Figure 1).

Rx Address Translator

The purpose of the Rx address translator 96 is to gather different (non-contiguous) words from a stream buffer 46 and to create a burst data to the PCI bus 24.

- 5 It is basically the inverse function of the address translation of the Tx address translator 54. The difference is that in this case a dynamic buffer structure must be considered. The burst data is transferred to the storage device 20 or the host 28 via
- 10 the PCI bus 24.

Figure 16 shows an example of an address translation applied to a incoming data stream to be stored in Disk 0, 1, 2 and 3 of the storage device 20. In this example, the following sequence of read addresses

15 for the stream buffer 48 is generated by the Rx address translator 96 to create a burst data 600.

- 178, 182, 13, 17, 1099, 1103 (for Disk 0)
 179, 183, 14, 18, 1100, 1104 (for Disk 1)
 20 180, 184, 15, 19, 1101, 1105 (for Disk 2)
 181, 185, 16... (for Disk 3)

CLAIMS

1. A method for translating a VPI/VCI of an ATM cell into an internal ID comprising the steps of:
- 5 distributing VPI/VCI entries into sections in a table according to a portion of each VPI/VCI entry;
- receiving an ATM cell;
- selecting a section to be searched according to the portion of a VPI/VCI of the received ATM cell;
- 10 performing a search over the selected section to find an entry corresponding to the VPI/VCI of the received ATM cell; and
- outputting an internal ID corresponding to the found entry.
- 15 2. A method according to claim 1, wherein the portion comprises the lower X bits of the VCI.
3. A method according to claim 1, wherein the entries are stored in an ordered way within each section and the search is a binary search.
- 20 4. An apparatus for translating a VPI/VCI of an ATM cell into an internal ID comprising:
- a table for storing VPI/VCI entries and being divided into sections;
- means for distributing VPI/VCI entries into the
- 25 sections in the table according to a portion of each VPI/VCI entry;
- means for selecting a section to be searched according to the portion of a VPI/VCI of an received ATM cell; and
- 30 means for performing a search over the selected section to find an entry corresponding to the VPI/VCI of the received ATM cell and for outputting an internal ID corresponding to the found entry.
5. An apparatus according to claim 4, wherein
- 35 the portion comprises the is lower X bits of the VCI.
6. An apparatus according to claim 4, wherein the entries are stored in an ordered way within each section and the search is a binary search.

7. An apparatus for sending data to an ATM network and receiving data from the ATM network comprising:

- 5 (a) a bus interface for interfacing with a bus supporting communication between a host, a storage device and the apparatus;
- (b) an ATM interface for interfacing with the ATM network;
- 10 (c) a transmission unit for transmitting outgoing data from the bus interface to the ATM interface, the transmission unit including
 - (1) a first RAM interface for interfacing with RAM being used as a buffer for buffering the outgoing data from the bus
 - 15 interface,
 - (2) means for segmenting the outgoing data from the buffer into outgoing ATM cells, and
 - (3) a traffic shaper, for controlling traffic of the outgoing ATM cells to the
 - 20 ATM interface in cooperation with the means for segmenting; and
- (d) a reception unit for transmitting incoming data from the ATM interface to the bus
- 25 interface, the reception unit including
 - (1) means for performing VPI/VCI filtering of incoming ATM cells,
 - (2) means for reassembling the incoming data using payload of the incoming ATM
 - 30 cells, and
 - (3) a second RAM interface for interfacing RAM being used as a buffer for buffering the incoming data the means for reassembling.

- 35 8. An apparatus according to claim 7, wherein the transmission unit comprises a command block for storing at least a pair of an address and a command, the address and the command being loaded from the host, and

detecting a match between the address and an address specified by a read pointer of the buffer, the command being executed when the match is detected.

9. An apparatus according to claim 8, wherein
5 the traffic shaper changes a stream bandwidth of the outgoing ATM cells according to the command.

10. An apparatus according to claim 8 or 9,
wherein the transmission unit comprises the means for inserting a Rate Change Indicator into the outgoing data
10 read from the buffer according to the command.

11. An apparatus according to claim 8, 9
wherein the or 4 command causes an interrupt of the host.

12. An apparatus according to any of claims 8-
11, wherein the transmission unit includes means for
15 swapping bytes of the outgoing data read from the buffer according to the command.

13. An apparatus according to any of claims 8-
12, wherein the means for segmenting applies AAL
segmentation with PDUs to the outgoing data read from the
20 buffer and changes the PDU-size according to the command.

14. An apparatus according to any of claims 7-
13, wherein the transmission unit includes an address
translator for providing write addresses for a burst data
from the storage device, at least a portion of the write
25 addresses being non-contiguous, and the buffer stores the burst data according to the write addresses and outputs data therefrom in a linear fashion.

15. An apparatus according to claim 14, wherein
the burst data includes payload for a predetermined
30 protocol, and the address translator receives packetisation parameters including at least size of the payload from the host and generates the write addresses so that a space for a protocol header of the predetermined protocol is reserved in the buffer
35 according to the packetisation parameters, and the buffer receives the protocol header in a burst from the host via the bus and stores the protocol header in the reserved space.

16. An apparatus according to claim 15, wherein the packetisation parameters include size of the protocol header.

17. An apparatus according to claim 15 or 16,
5 wherein the predetermined protocol is TCP(or UDP) and/or IP.

18. An apparatus according to claim 15, 16 or 17, wherein the transmission unit comprises means for receiving a partial checksum calculated by the host,
10 updating the partial checksum to obtain a complete checksum and filling the complete checksum into the protocol header in the buffer when the protocol header and the payload have been written in the buffer.

19. An apparatus according to any of claims 7-
15 18, wherein the traffic shaper comprises:

means for classifying one or more first streams of the ATM cells into one or more classes, each class including one or more streams having the same bit rate characteristics; storage means for storing a set of
20 parameters to control the bit rate for each class; and means for executing a rate pacing of each class according to the set of parameters in the storage means.

20. An apparatus according to claim 19, wherein the parameters include an ideal scheduled time and an
25 increment for the ideal scheduled time, and the means for executing increments the ideal scheduled time with the increment when a ATM cell of a current stream in a class is put into a transmit queue, compares a reference clock with the incremented ideal scheduled time and puts a ATM
30 cell of a next stream in the class into the transmit queue when the incremented ideal scheduled time becomes equal to or less than the reference clock.

21. An apparatus according to claim 20, wherein the increment of each class takes two alternative values,
35 one of the values indicating a bit rate higher than a required bit rate and the other indicating a bit rate lower than the required bit rate, and the means for

executing alternates the values of the increment cyclically.

22. An apparatus according to claim 19, 20 or 21, wherein the increment takes a value so that the cells of the streams belonging to the same class are evenly spaced.

23. An apparatus according to any of claims 19-22, wherein the first streams have high priority, and the traffic shaper comprises means for shaping traffic of one or more second streams of the ATM cells having low priority with leaky bucket mechanism.

24. An apparatus according to any of claims 19-23, wherein the first streams include a VBR stream, and when a bit rate characteristic of the VBR stream changes, the means for classifying detaches the VBR stream from a current class and attaches the VBR stream to a new class, and the parameters of the current class and the new class are updated.

25. An apparatus according to any of claims 7-24, wherein the means for performing VPI/VCI filtering comprises:

a table for storing VPI/VCI entries and being divided into sections;

means for distributing VPI/VCI entries into the sections in the table according to a portion of each VPI/VCI entry;

means for selecting a section to be searched according to the portion of a VPI/VCI of a received ATM cell; and means for performing a search over the selected section to find an entry corresponding to the VPI/VCI of the received ATM cell and outputting an internal ID corresponding to the found entry.

26. An apparatus according to claim 25, wherein the portion comprise the lower X bits of the VCI.

27. An apparatus according to claim 25 or 26, wherein the entries are stored in an ordered way within each section and the search is a binary search.

28. An apparatus according to any of claims 7-27, wherein the reception unit includes means for verifying a checksum in a protocol header of the incoming data.

5 29. An apparatus according to any of claims 7-28, wherein the reception unit includes an address translator for gathering non-contiguous words from the buffer to create a burst data for the bus interface.

10 30. A system comprising an apparatus according to any of claims 7-29, the host for running applications and managing delivery of data, the storage device for storing the incoming data and retrieving the stored incoming data as the outgoing data, and the bus.

15 31. System according to claim 30, wherein the storage device includes a RAID or RDI system, and the transmission unit includes a RAID function for correcting error in the outgoing data using RAID redundancy and changing whether performing an error correction or not according to the command.

20 32. A system according to claim 30 or 31, wherein the storage device includes a RAID or RDI system, and adjacent words in the burst data belong to different stripe units.

25 33. A system according to claim 30, 31 or 32, wherein the reception unit includes:

 a pattern detector for detecting at least a preset bit pattern in the incoming data when the incoming data is transmitted from the ATM interface to the bus interface; and

30 a list for storing location information corresponding to a location of the preset bit pattern in the data when the present bit pattern is detected by the pattern detector;

35 the host controlling a delivery of the data from the storage device to the network according to the location information in the list.

 34. A system according to claim 33, wherein the incoming data includes video data compressed according to

MPEG standard and the location information represents at least a location of I-picture, and the host controls the storage device so that the I-picture is accessed according to the location information.

- 5 35. An apparatus according to any of claims 30-34, wherein the storage device includes a RAID or RDI system, and the reception unit includes means for adding RAID redundancy to the incoming data.

- 10 36. A method for sending/receiving data to/from an ATM network, using an apparatus according to any of claims 7-29 and/or a system according to any of claims 30-35.

37. A method for streaming data from a storage device comprising the steps of:

- 15 providing write addresses for a burst data to a buffer, at least a portion of the write addresses being non-contiguous;

- transferring the burst data from the storage device to the buffer via a bus supporting communication
20 between a host, the storage device and a streaming device;

writing the burst data in the buffer according to the write addresses; and

- 25 reading data from the buffer in a linear fashion.

38. A method according to claim 37, wherein the burst data includes payload for a predetermined protocol, and further comprises:

- 30 receiving packetisation parameters including at least a size of the payload from the host;

generating the write addresses so that a space for protocol header of the predetermined protocol is reserved in the buffer according to the packetisation parameters;

- 35 transferring the protocol header in a burst from the host to the buffer via the bus; and

writing the protocol header in the reserved space in the buffer.

39. A method according to claim 38, wherein the packetisation parameters include a size of the protocol header.

40. A method according to claim 38, wherein the
5 predetermined protocol is TCP (or UDP) and/or IP.

41. A method according to claim 38, further comprises:

receiving a partial checksum calculated by the host;

10 updating the partial checksum to obtain a complete checksum; and

filling the complete checksum into the protocol header in the buffer when the protocol header and the payload have been written in the buffer.

15 42. A method according to claim 37, wherein the storage device includes a RAID or RDI system, and adjacent words in the burst data belong to different stripe units.

43. A streaming device for streaming a data
20 from a storage device comprising:

means for receiving a burst data from the storage device via a bus supporting communication between a host, the storage device and the streaming device;

25 means for providing write addresses for the burst data, at least a portion of the write addresses being non-contiguous; and

a buffer for storing the burst data according to the write addresses and outputting data therefrom in a linear fashion.

30 44. A streaming device according to claim 43, wherein the burst data includes payload for a predetermined protocol, and the means for providing receive packetisation parameters including at least a size of the payload from the host and generate the write
35 addresses so that a space for a protocol header of the predetermined protocol is reserved in the buffer according to the packetisation parameters, and the buffer receives the protocol header in a burst from the host via

the bus and stores the protocol header in the reserved space.

45. A device according to claim 44, wherein the packetisation parameters include size of the protocol header.

46. A device according to claim 44, wherein the predetermined protocol is TCP(or UDP) and/or IP.

47. A device according to claim 44, further comprises means for receiving a partial checksum
10 calculated by the host, updating the partial checksum to obtain a complete checksum and filling the complete checksum into the protocol header in the buffer when the protocol header and the payload have been written in the buffer.

15 48. A device according to claim 43, wherein the storage device includes a RAID or RDI system, and adjacent words in the burst data belong to different stripe units.

49. A method for delivering data comprising the
20 steps of:

loading at least a pair of an address and a command from a host;

storing the data in a buffer;

reading the data from the buffer according to a
25 read pointer;

executing the command if a match between the address and an address specified by the read pointer is detected; and

delivering the data read from the buffer after
30 the execution of the command.

50. A method according to claim 49, wherein the command causes changing a stream bandwidth of the data read from the buffer.

51. A method according to claim 49, wherein the
35 command causes inserting a Rate Change Indicator into the data read from the buffer.

52. A method according to claim 49, wherein the command causes an interrupt of the host.

53. A method according to claim 49, wherein the data is read from a RAID or RDI system and is provided to the buffer and the command enables and disables a RAID function for correcting an error.

5 54. A method according to claim 49, wherein the command causes swapping bytes of the data read from the buffer.

55. A method according to claim 49, wherein the command causes changing a PDU-size in AAL segmentation
10 applied to the data read from the buffer.

56. A device for delivering data comprising:
a command block for storing at least a pair of
an address and a command loaded from a host and detecting
a match between the address and an address specified by a
15 read pointer of a buffer buffering the data;

means for executing the command in cooperation
with the command block when the match is detected; and

means for delivering the data read from the
buffer after the execution of the command.

20 57. A device according to claim 56, wherein the means for executing changes of a stream bandwidth of the data read from the buffer according to the command.

58. A device according to claim 56, wherein the means for executing inserts a Rate Change Indicator into
25 the data read from the buffer according to the command.

59. A device according to claim 56, wherein the command causes an interrupt of the host.

60. A device according to claim 56, wherein the data is read from a RAID or RDI system and is provided to
30 the buffer, and the means for executing changes whether performing an error correction or not according to the command.

61. A device according to claim 56, wherein the means for executing swaps bytes of the data read from the
35 buffer according to the command.

62. A device according to claim 56, wherein the means for executing applies AAL segmentation with PDUs to

the data read from the buffer and changes the PDU-size according to the command.

63. A method for delivering data comprising the steps of:

- 5 receiving data from a network;
- detecting at least a preset bit pattern in the data when the received data is transmitted to a storage device;
- adding location information corresponding to a
- 10 location of the preset bit pattern in the data to a list when the preset bit pattern is detected;
- storing the data in the storage device; and
- controlling a delivery of the data from the storage device to the network according to the location
- 15 information in the list.

64. A method according to claim 63, wherein the data includes video data compressed according to MPEG standard and the location information represents at least a location of I-picture, and the step of controlling

20 comprises:

 accessing the I-picture of the data stored in the storage device according to the location information.

65. An apparatus for delivering data comprising:

- 25 receiving means for receiving data from a network;
- a pattern detector for detecting at least a preset bit pattern in the data when the data is transmitted from the receiving means to a storage device
- 30 storing the data;
- a list for storing location information corresponding to a location of the preset bit pattern in the data when the preset bit pattern is detected by the pattern detector; and
- 35 means for controlling a delivery of the data from the storage device to the network according to the location information in the list.

66. An apparatus according to claim 65, wherein the data includes video data compressed according to MPEG standard and the location information represents at least a location of I-picture, and the means for controlling
5 controls the storage device so that the I-picture is accessed according to the location information.

67. A traffic shaping method comprising the steps of:

classifying one or more first streams into one
10 or more classes, each class including one or more streams having the same bit rate characteristics;

setting a set of parameters to control the bit rate for each class; and

executing a rate pacing of each class according
15 to the set of parameters.

68. A traffic shaping method according to claim 67, wherein the parameters include an ideal scheduled time and an increment for the ideal scheduled time, and the step of executing comprises:

20 incrementing the ideal scheduled time with the increment when a cell of a current stream in a class is put into a transmit queue;

comparing a reference clock with the incremented ideal scheduled time; and

25 putting a cell of a next stream in the class into the transmit queue when the incremented ideal scheduled time becomes equal to or less than the reference clock.

69. A traffic shaping method according to claim 30 68, wherein the increment of each class takes two alternative values, one of the values indicating a bit rate higher than a required bit rate and the other indicating a bit rate lower than the required bit rate, and the step of executing comprises:

35 alternating the values of the increment cyclically.

70. A traffic shaping method according to claim 68, wherein the increment takes a value so that the cells

of the streams belonging to the same class are evenly spaced.

71. A traffic shaping method according to claim 67, wherein the first streams have high priority, and
5 further comprises:

shaping traffic of one or more second streams having low priority with leaky bucket mechanism.

72. A traffic shaping method according to claim 67, wherein the first streams include a VBR stream, and
10 further comprises:

detaching the VBR stream from a current class when a bit rate characteristic of the VBR stream changes;
updating the parameters of the current class;
attaching the VBR stream to a new class; and
15 updating the parameters of the new class.

73. A traffic shaper comprising:
means for classifying one or more first streams into one or more classes, each class including one or more streams having the same bit rate characteristics;
20 storage means for storing a set of parameters to control the bit rate for each class; and
means for executing a rate pacing of each class according to the set of parameters in the storage means.

74. A traffic shaper according to claim 73,
25 wherein the parameters include an ideal scheduled time and an increment for the ideal scheduled time, and the means for executing increments the ideal scheduled time with the increment when a cell of a current stream in a class is put into a transmit queue, compares a reference
30 clock with the incremented ideal scheduled time and puts a cell of a next stream in the class into the transmit queue when the incremented ideal scheduled time becomes equal to or less than the reference clock.

75. A traffic shaper according to claim 74,
35 wherein the increment of each class takes two alternative values, one of the values indicating a bit rate higher than a required bit rate and the other indicating a bit rate lower than the required bit rate, and the means for

executing alternates the values of the increment cyclically.

76. A traffic shaper according to claim 74, wherein the increment takes a value so that the cells of
5 the streams belonging to the same class are evenly spaced.

77. A traffic shaper according to claim 73, wherein the first streams have high priority, and further comprises means for shaping traffic of one or more second
10 streams having low priority with leaky bucket mechanism.

78. A traffic shaper according to claim 73, wherein the first streams include a VBR stream, and, when a bit rate characteristic of the VBR stream changes, the means for classifying detaches the VBR stream from a
15 current class and attaches the VBR stream to a new class, and the parameters of the current class and the new class are updated.

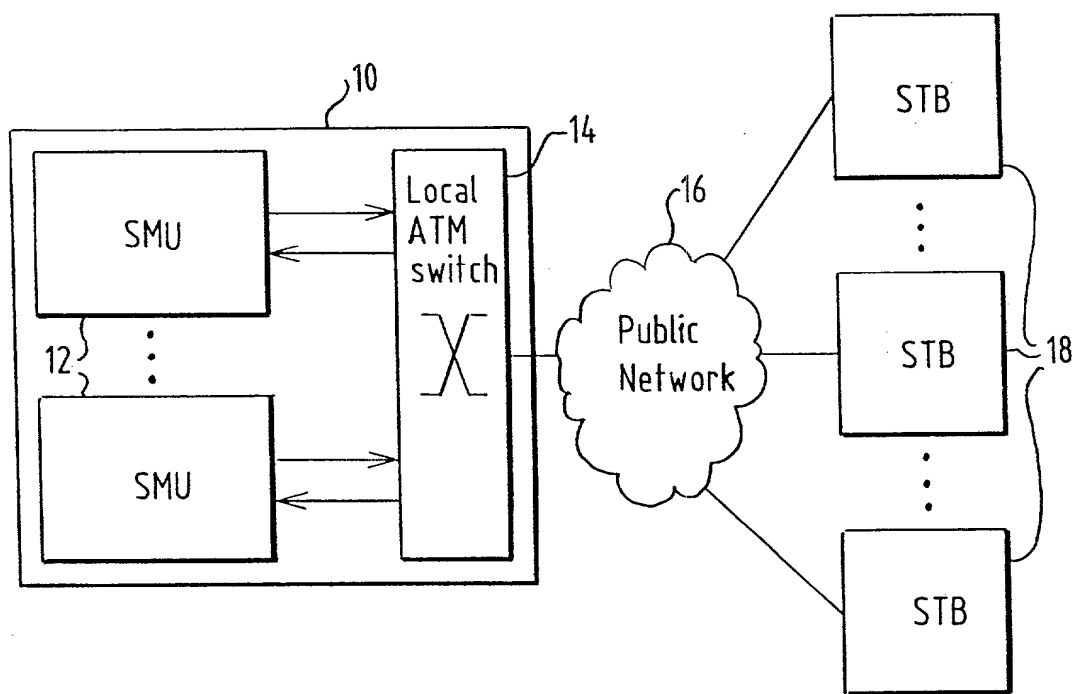


FIG.1

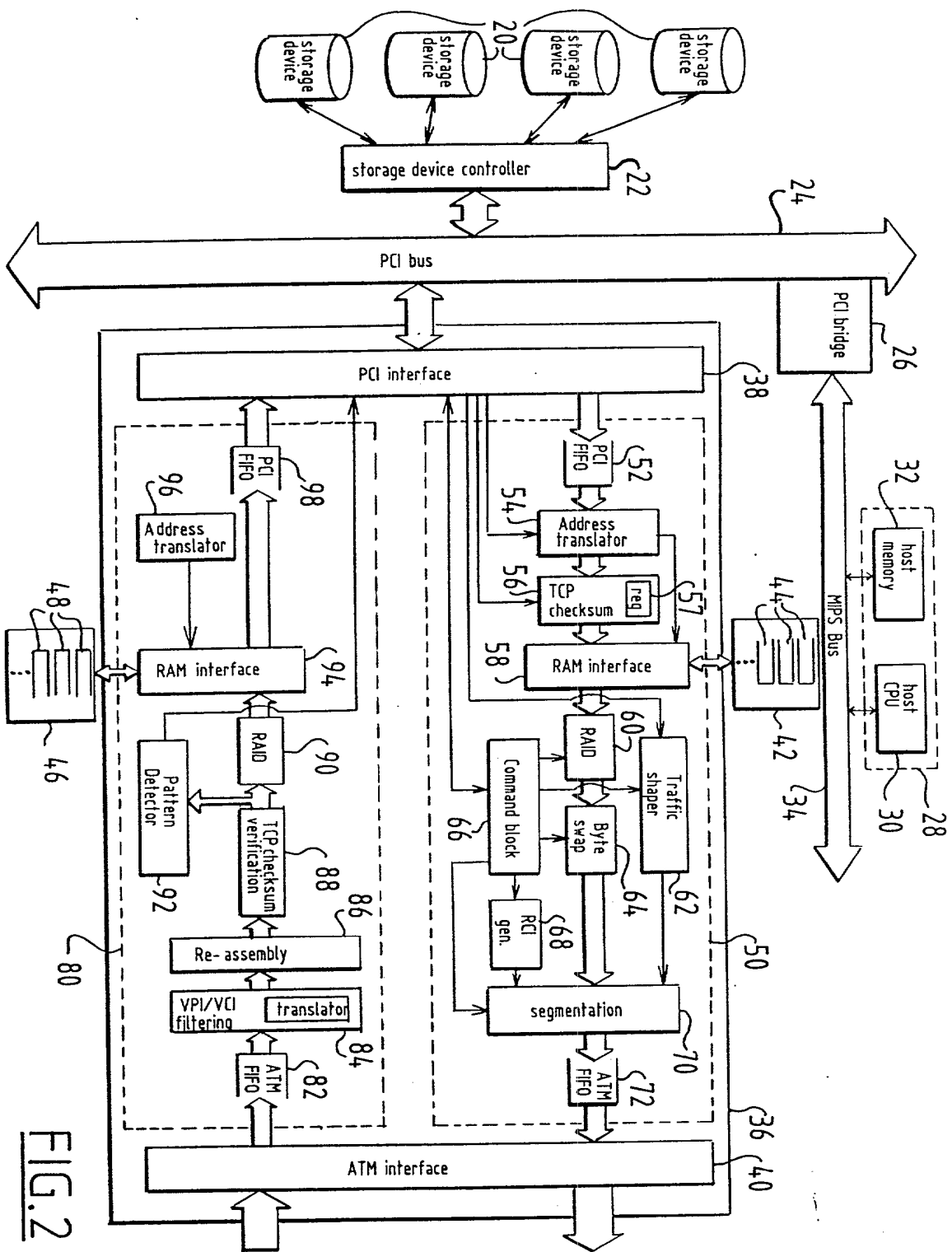


FIG. 2

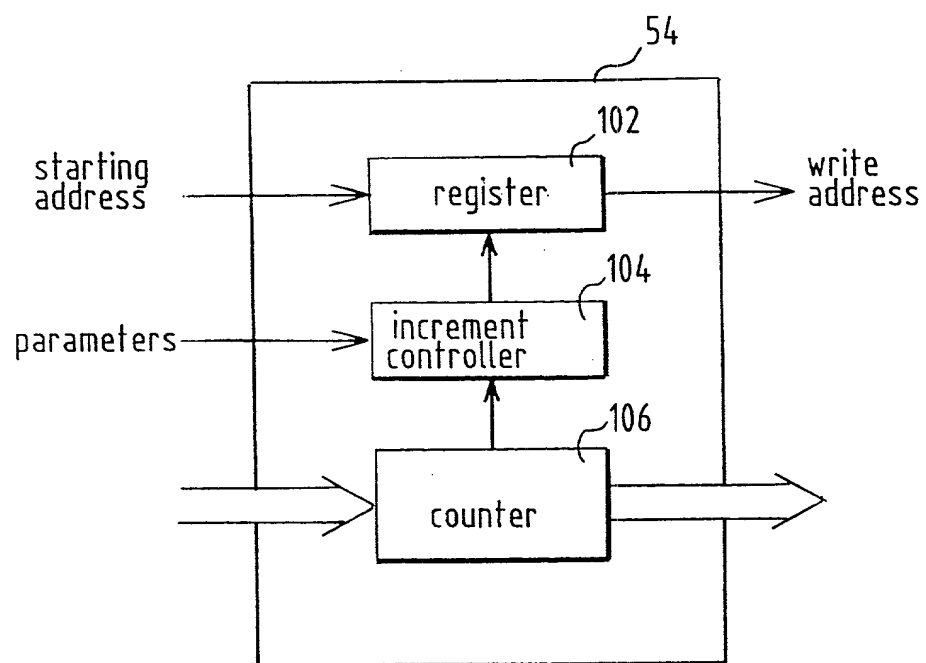
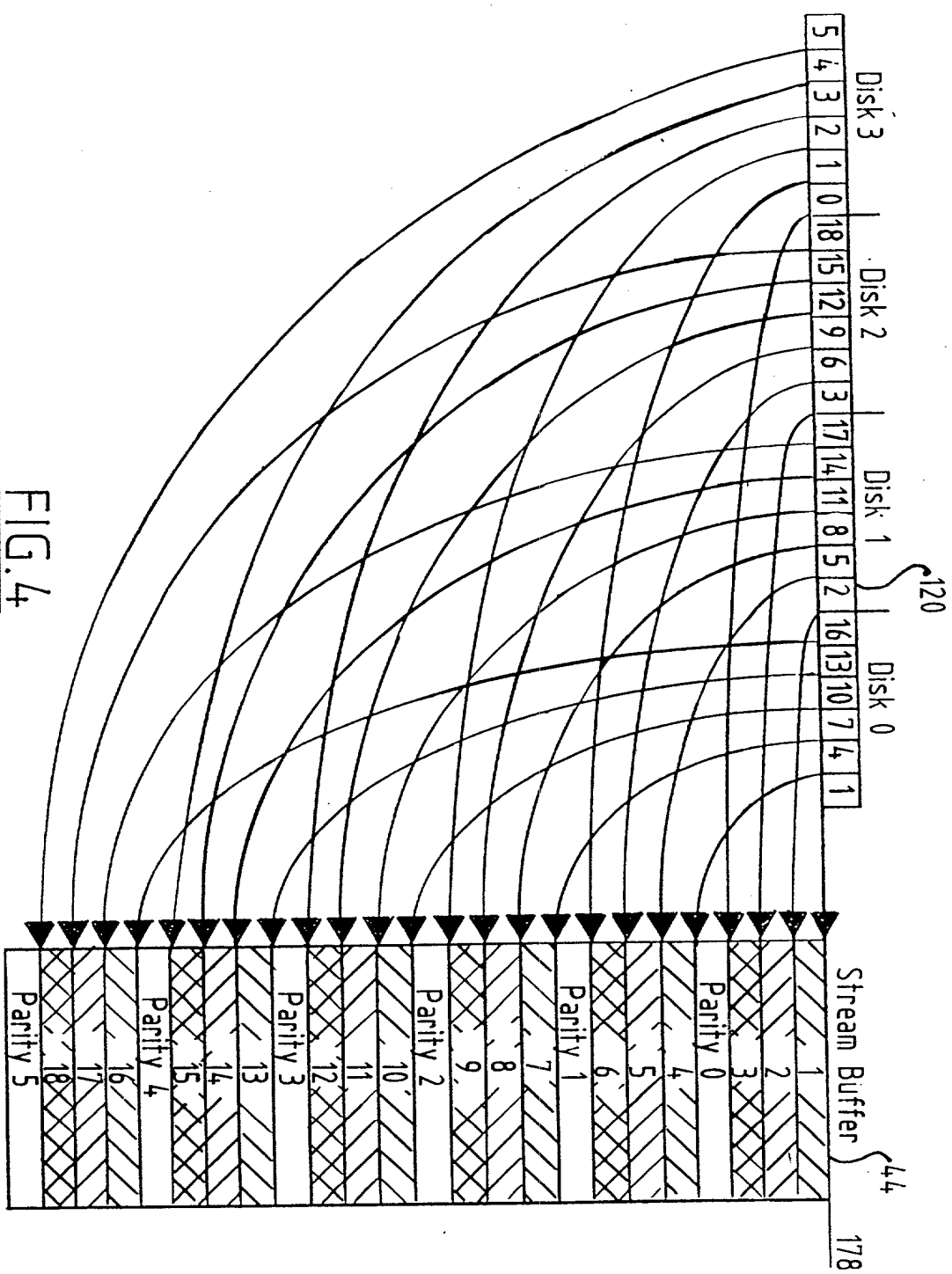
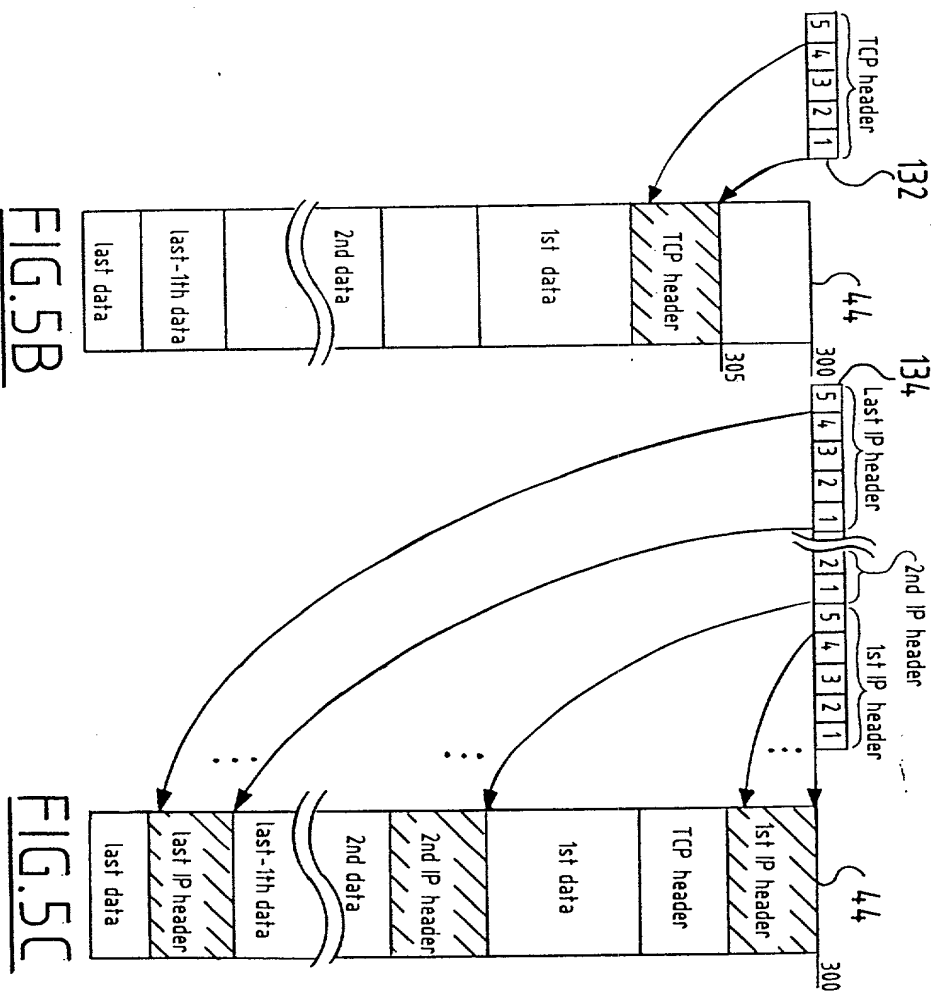
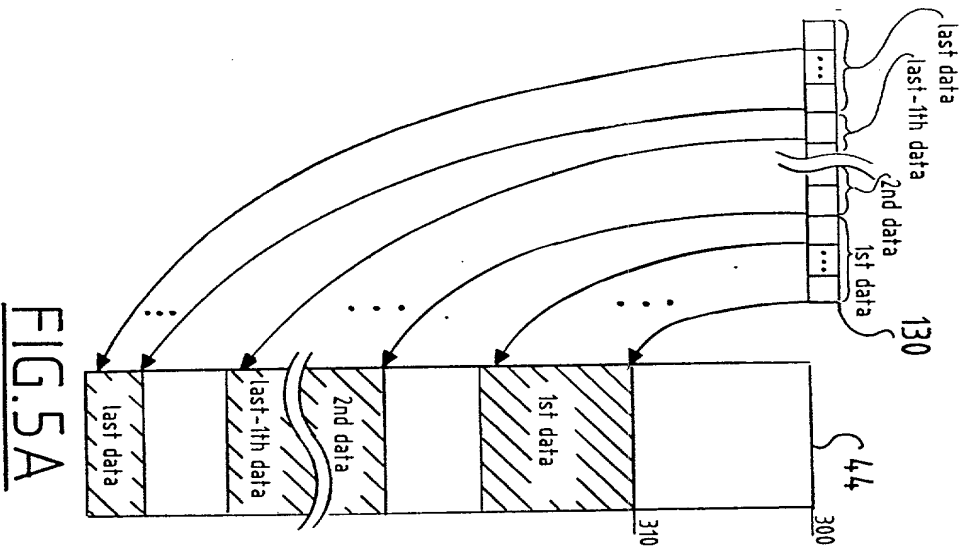
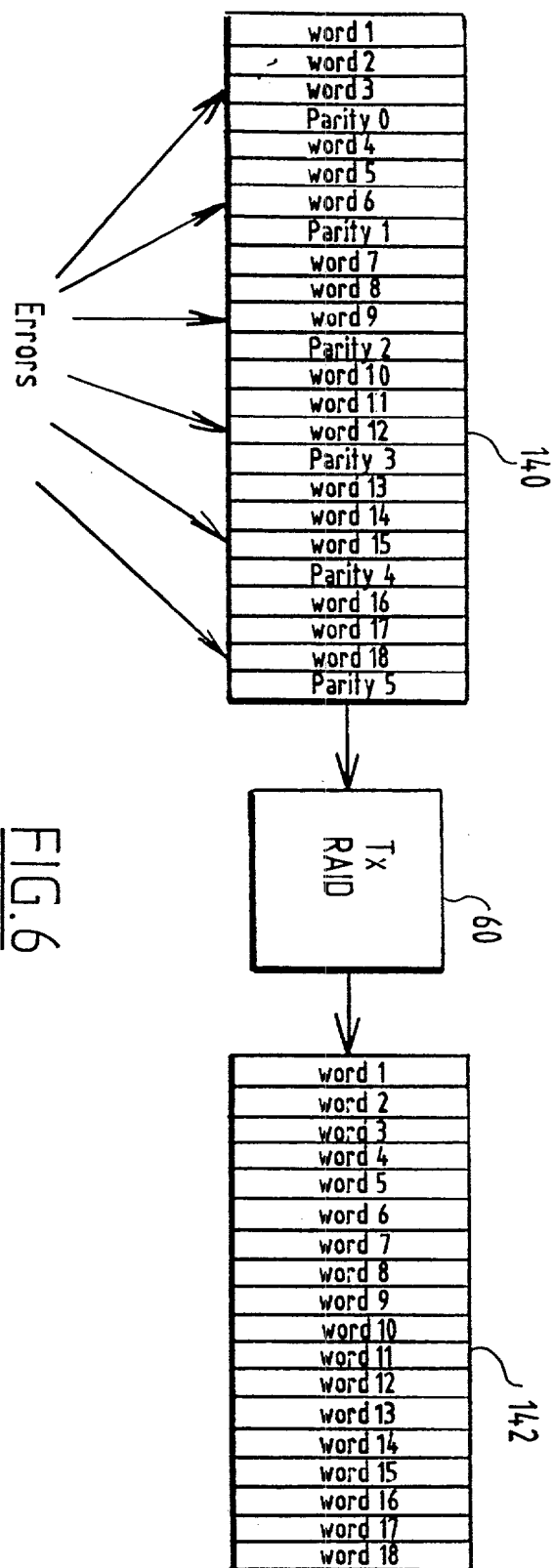


FIG.3







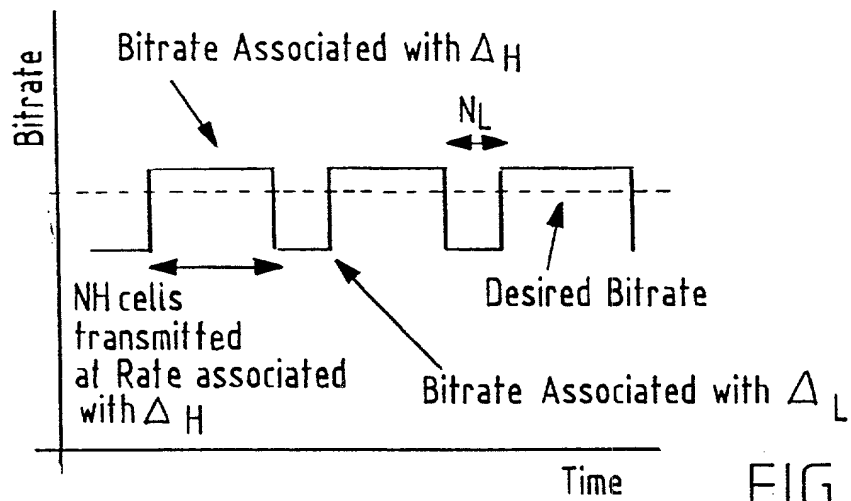


FIG. 7

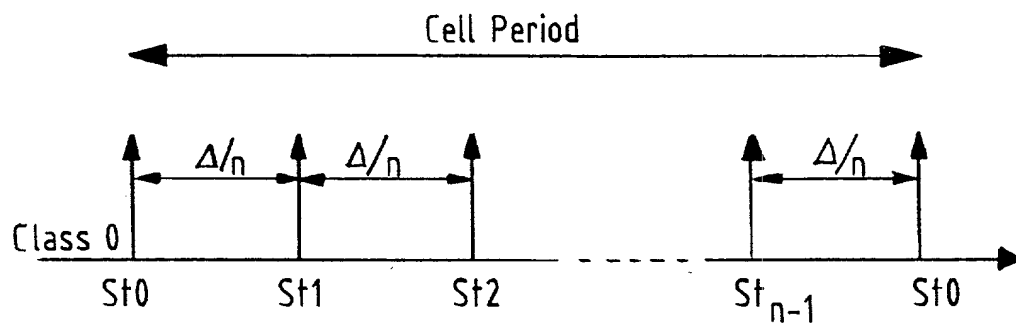


FIG. 8

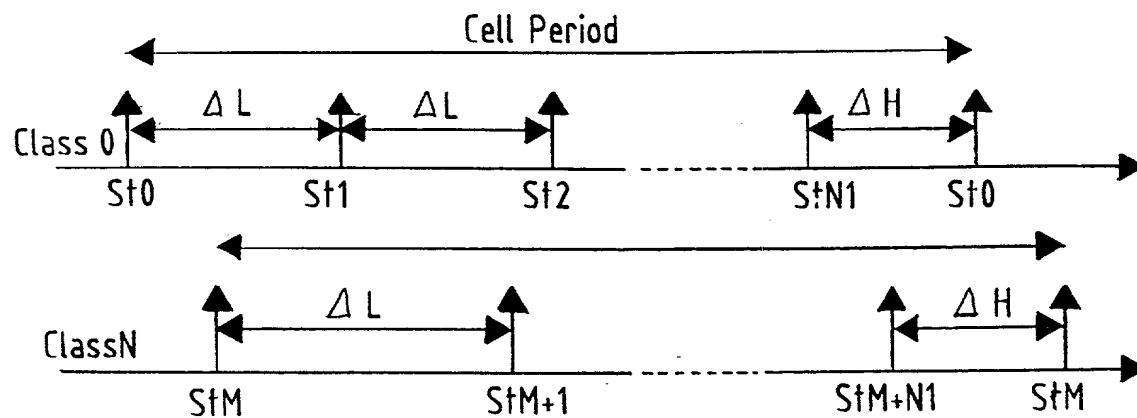


FIG. 9

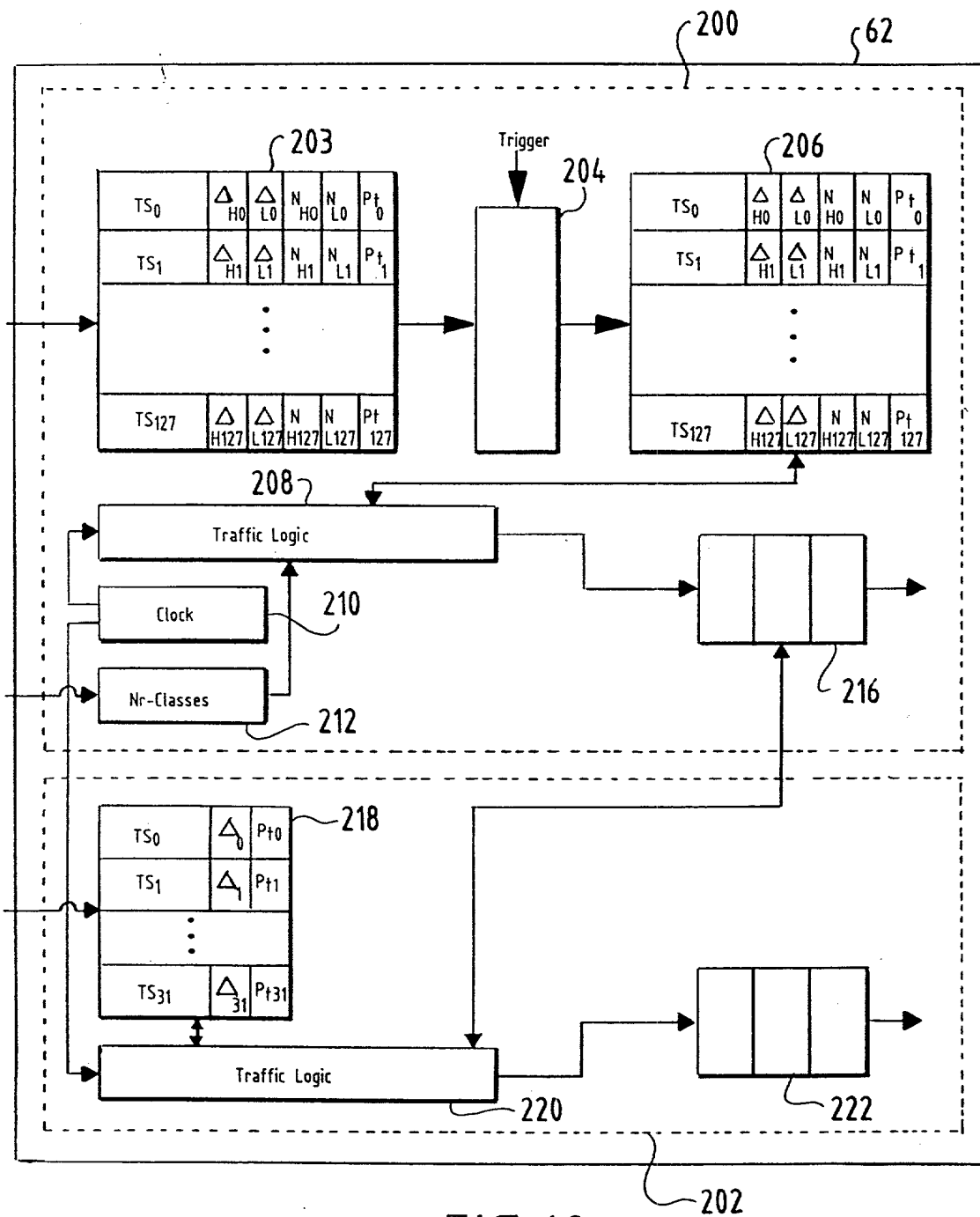


FIG. 10

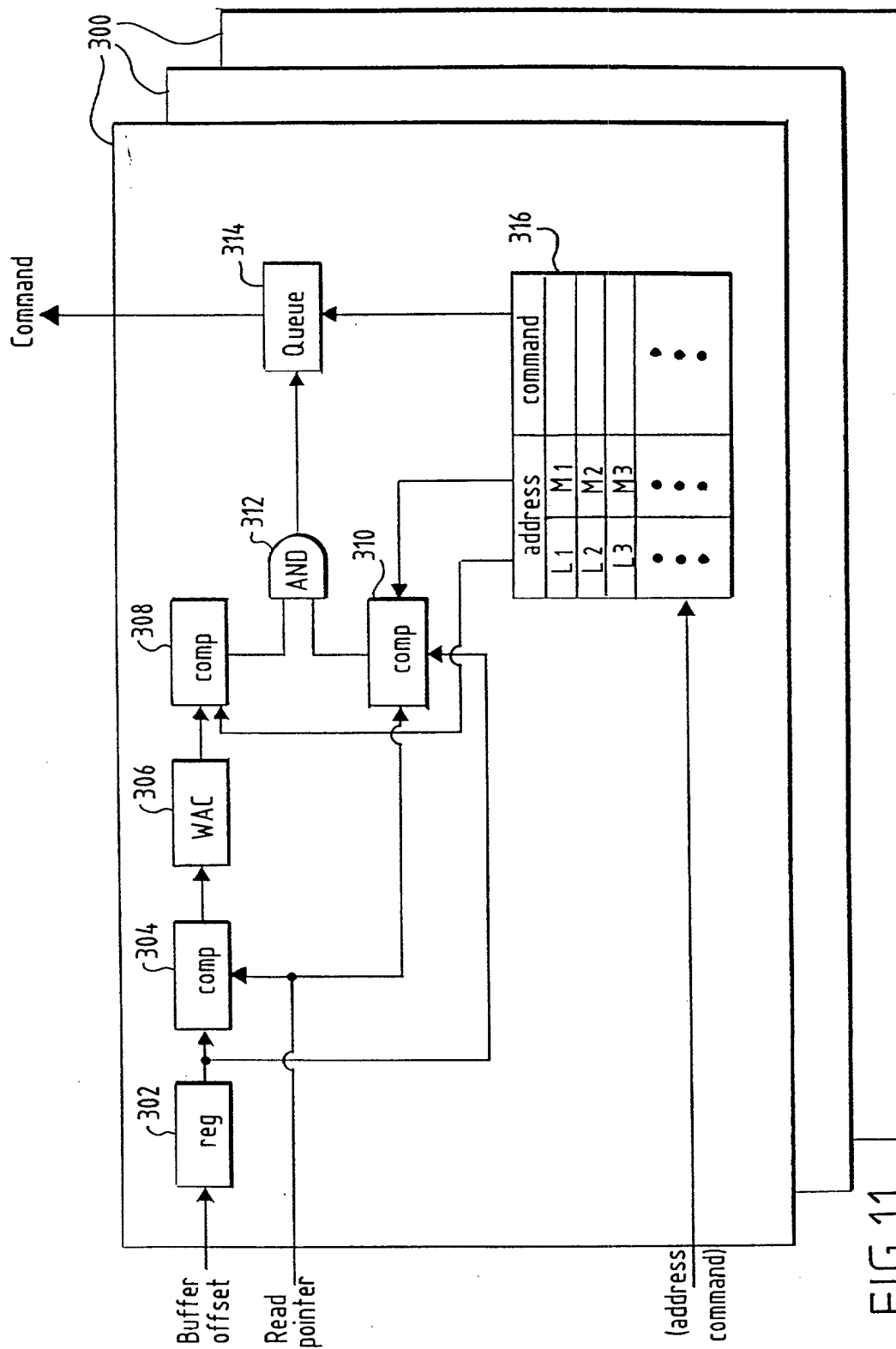


FIG. 11

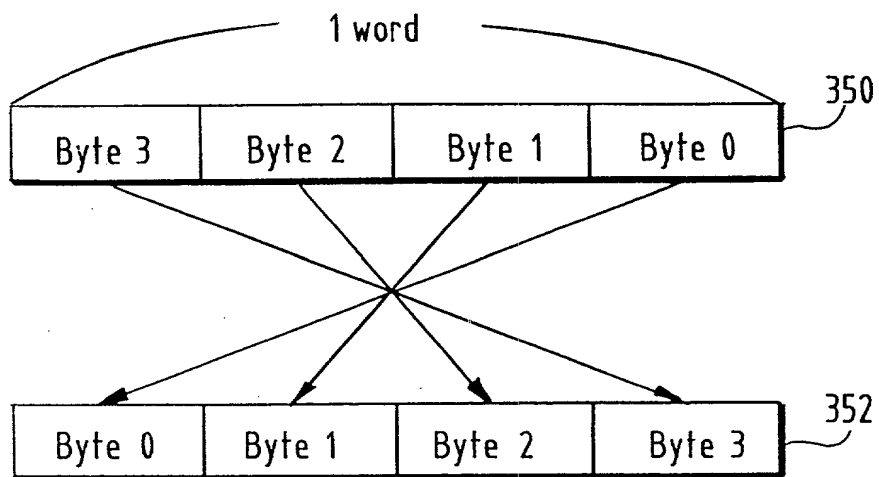


FIG.12

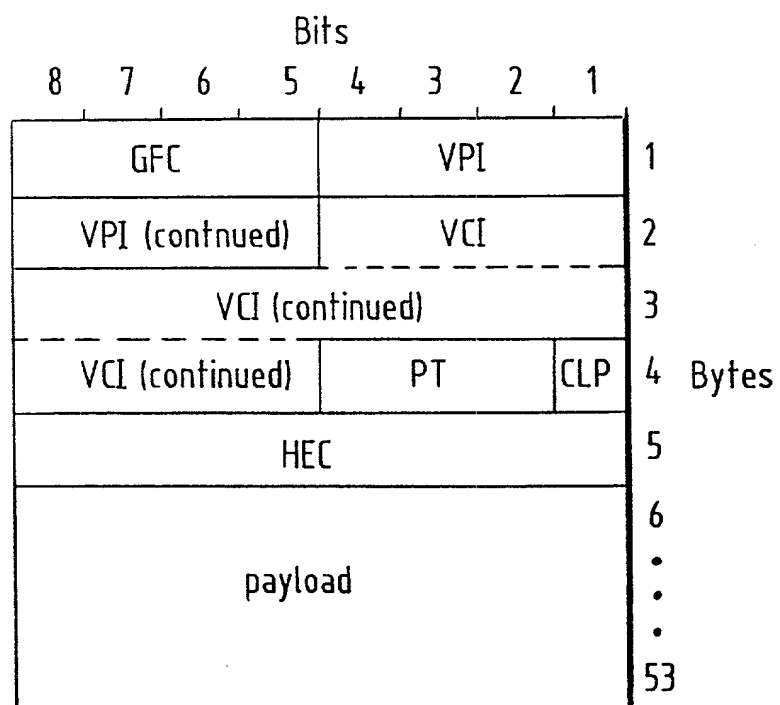


FIG.13

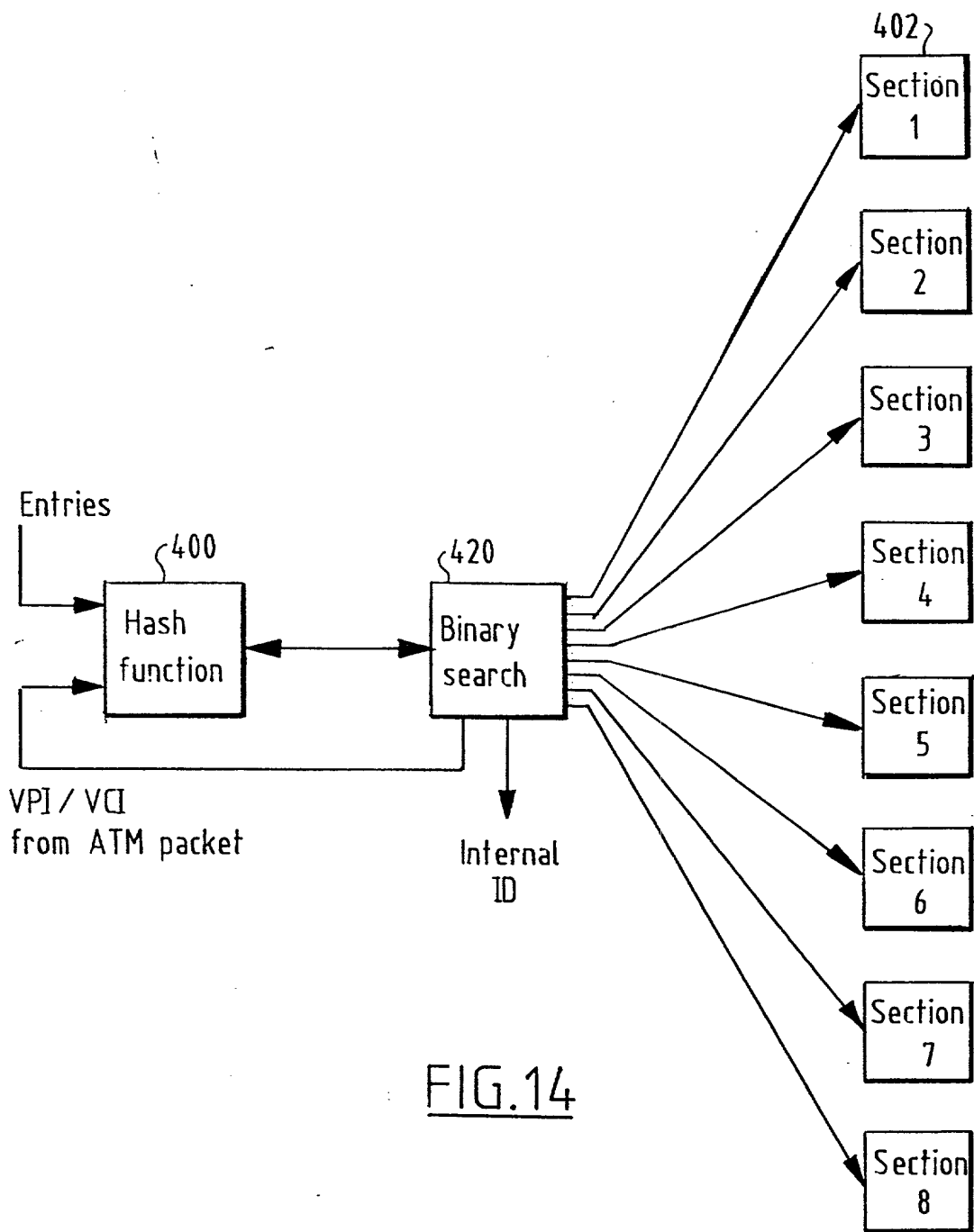


FIG.14

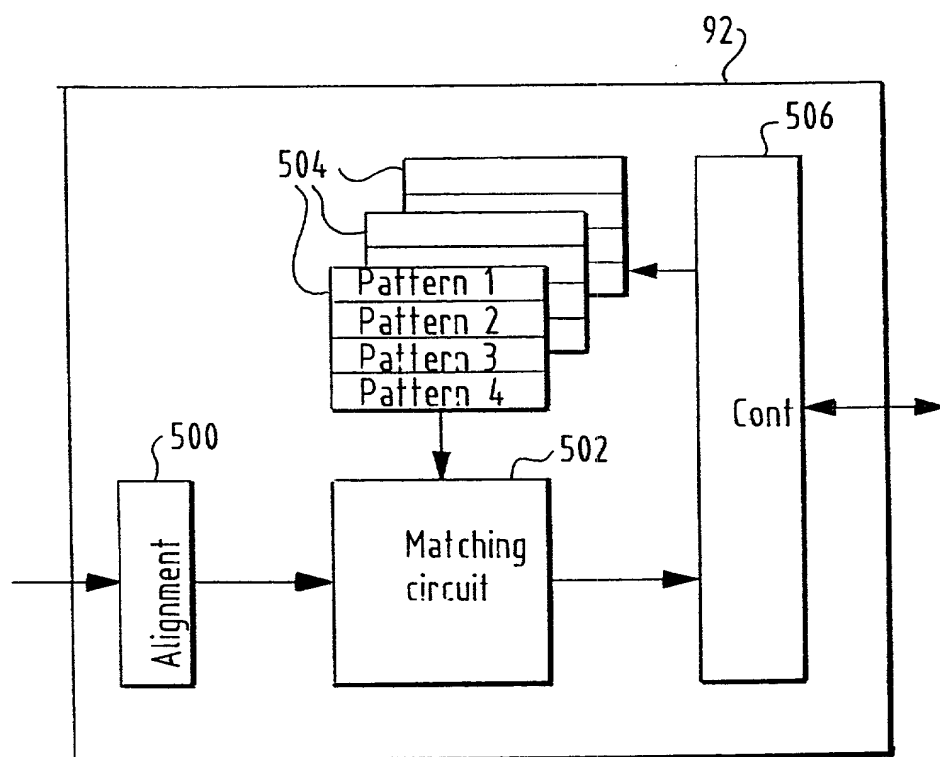


FIG.15

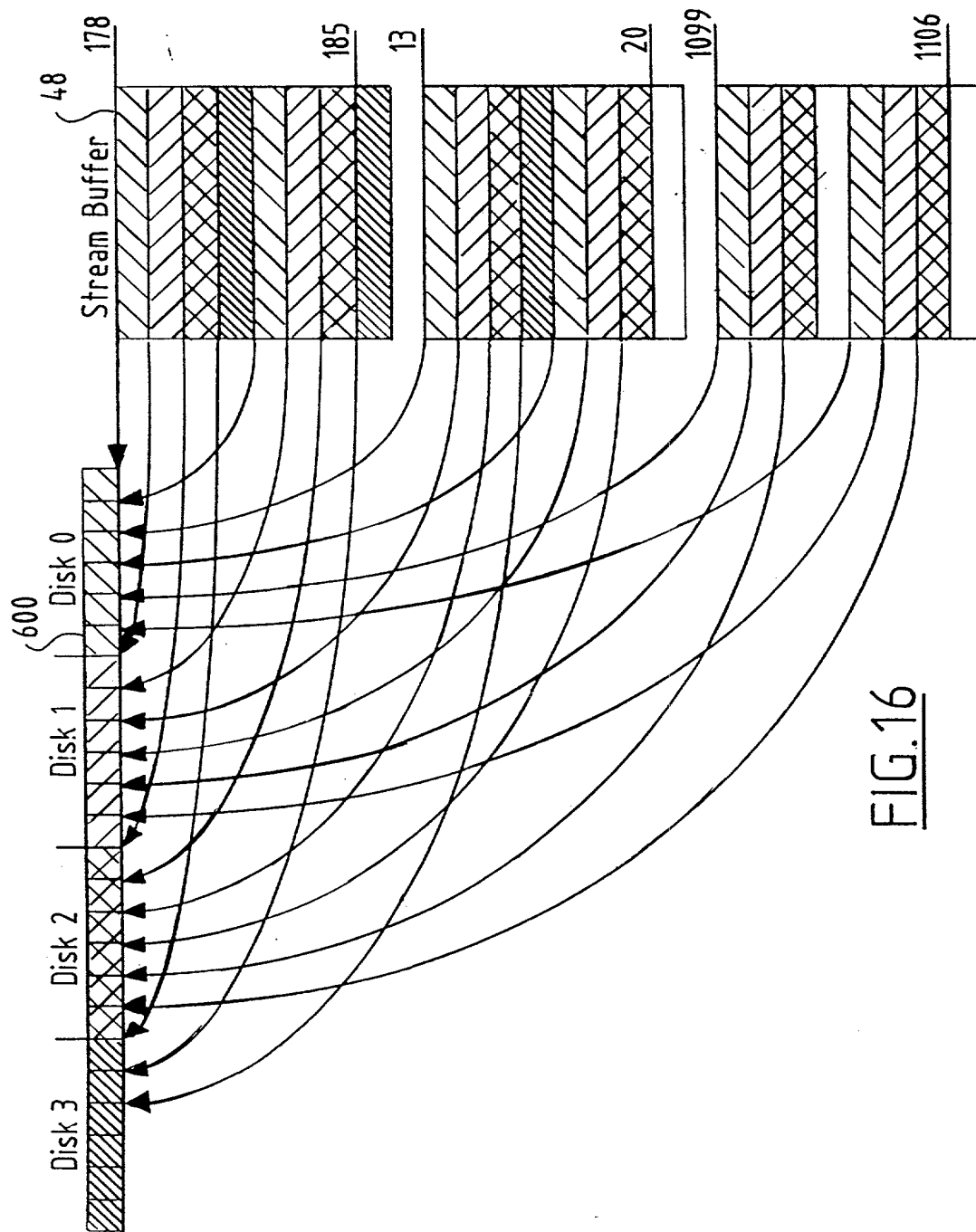


FIG.16

DECLARATION FOR PATENT APPLICATION (JOINT OR SOLE)

(Under 37 CFR § 1.63; with Power of Attorney)

FROMMER LAWRENCE & HAUG LLP

FLH File No. 450117-4278

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention ENTITLED:

METHOD AND APPARATUS FOR SERVING DATA

the specification of which

_____ is attached hereto.

☒ was filed on _____ as Application Serial No. _____,

with amendment(s) through _____ (if applicable, give dates).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Sec. 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>	<u>List additional applications on separate page:</u>	<u>Priority Claimed:</u>		
<u>Number:</u>	<u>Country:</u>	<u>Filed (Day/Month/Year):</u>	<u>Yes</u>	<u>No</u>
96203338.7	EP	27 November 1996	X	
96203341.1	EP	27 November 1996	X	
96203340.3	EP	27 November 1996	X	
96203339.5	EP	27 November 1996	X	
96203334.6	EP	27 November 1996	X	
96203336.1	EP	27 November 1996	X	

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code § 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Sec. 1.56, which became available between the filing date of the prior application and the national or PCT international filing date of this application:

Prior U.S. Application(s) (list additional applications on separate page):

Appln. Ser. Number: _____ Filed (Day/Month/Year): _____ Status (patented, pending, abandoned): _____

I hereby appoint WILLIAM S. FROMMER, Registration No. 25,506, or his duly appointed associate, my attorneys, with full power of substitution and revocation, to prosecute this application, to make alterations and amendments therein, to file continuation and divisional applications thereof, to receive the Patent, and to transact all business in the Patent and Trademark Office and in the Courts in connection therewith, and specify that all communications about the application are to be directed to the following correspondence address:

WILLIAM S. FROMMER, Esq.
c/o FROMMER LAWRENCE & HAUG LLP
745 Fifth Avenue
New York, New York 10151

Direct all telephone calls to:
(212) 588-0800
to the attention of:
WILLIAM S. FROMMER

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

INVENTOR(S):

Signature: _____

Full name of sole or first inventor:

Residence:

Citizenship:

PAUL HODGINS

Brussels, Belgium

Canada

Date: 14 Jan 1998

Signature: _____

Full name of 2nd joint inventor (if any):

Residence:

Citizenship:

GERT JOSEF ELISA COPEJANS

Brussels, Belgium

Belgium

Date: 12 Jan 1998

[Similarly list additional inventors on separate page]

Post Office Address(es) of inventor(s):

[If all inventors have the same post office address]

SONY EUROPA B.V.

Schipholweg 275

NL-1711 Pk Badhoevedorp

THE NETHERLANDS

Note: In order to qualify for reduced fees available to Small Entities, each inventor and any other individual or entity having rights to the invention must also sign an appropriate separate "Verified Statement (Declaration) Claiming [or Supporting a Claim by Another for] Small Entity Status" form [e.g. for Independent Inventor, Small Business Concern, Nonprofit Organization, individual Non-Inventor].

Note: A post office address must be provided for each inventor.

ADDITIONAL INVENTORS

FLH File No. 450117-4278

Signature: _____

Full name of 3rd joint inventor (if any):

YOERI APTS

Residence:

Brussels, Belgium

Citizenship:

Belgium

Date: _____

12/1/98

Signature: _____

Full name of 4th joint inventor (if any):

JOHAN DE VOS

Residence:

Brussels, Belgium

Citizenship:

Belgium

Date: _____

12/1/98